EXPERIMENTAL DESIGN AND MATHEMATICAL MODELING IN ODE BASED SYSTEMS BIOLOGY

A Dissertation
Presented to
The Academic Faculty

By

Jenny Eunice Jeong

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
School of Electrical and Computer Engineering

Georgia Institute of Technology

May 2019

EXPERIMENTAL DESIGN AND MATHEMATICAL MODELING IN ODE BASED SYSTEMS BIOLOGY

Approved by:

Dr. Peng Qiu
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Robert J. Butera
School of Electrical Engineering
Georgia Institute of Technology

Dr. Eberhard O. Voit
School of Biomedical Engineering
Georgia Institute of Technology

Dr. Fumin Zhang
School of Electrical Engineering
Georgia Institute of Technology

Dr. Omer T. Inan
School of Electrical Engineering
Georgia Institute of Technology

Dr. Eva L. Dyer
School of Biomedical Engineering
Georgia Institute of Technology

Date Approved: Nov. 8th, 2018

To my family and friends.

ACKNOWLEDGEMENTS

Fortunately, my PhD journey was not alone and filled with many people who kindly support me and willingly help me both physically and mentally. First and foremost, I wolud thank my Advisor Dr. Peng Qiu for insightful guidance, encouragements during my Ph.D studies. I would also like to thank all of the committee members, and the group members in Dr. Qiu's group, for sharing their knowledge and experience.

I would like to thank my friends and my family. My parents and my sister have been truly supportive and encouraging during my Ph.D studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

Studying biological behaviors is important in understanding complex process mechanisms. To understand and describe the biological processes, system-level approaches has been developed, known as Systems biology. Systems biology allows us to control and simulate the biological processes by formulating mathematical models representing the complex mechanism of the process. Especially, the ordinary differential equations (ODEs) model is usually used to model the biological processes because the ODE model can describe changes of systems components status over time. For example, the change of concentration of an enzyme can be expressed using the first derivative with respect to time. Typically, an ODE model contains many unknown model parameters, thus these parameters should be estimated based on the experimentally observed data. However, the amount of experimental data is almost always limited compared to the complexity of the model, and there is always information gap between the complexity of the model and available data. To overcome this gap, experimental design and model reduction methods have been developed. Experimental design methods increase the amount of data by suggesting the most informative experiment, and the model reduction method reduces the complexity of the model by finding the key controlling part of the model. In chapter 2, the possibility of unified framework of these two distinctive methods has been shown using a simple example model. If this unified method is developed, we can expect to improve parameter estimation by bridging the information gap between data and the model.

To improve parameter estimation without increasing data and reducing the complexity of the model, a new approach which quantifies the relative importance of each data point and gives a different weight to each data according to the quantified importance has been introduced in chapter 3. Each weight is quantified based on the difficulty of predicting this data point using the other data points. For instance, if one data point is located in a dynamically changing region, this should receive higher weight, while if one data point is

located in a steady state region, this data point will receive relatively lower weight. By giving different weights according to their relative importance, we can estimate parameters that generate more accurate simulation data in the dynamically changing regions.

From chapter 2 and 3, we can see the importance of mathematical modeling for studying the biological systems. In chapter 3, the way of formulating the ODE model will be discussed. Passive microfluidic cell sorting device is used as an application. This device can sort the cells based on a cell's biophysical properties, such as cell stiffness or size, for purpose of quantification or experiment. For example, since the trajectories of stiff cells and soft cells are different, we can sort the cells based on the stiffness of the cells. The data for this model can be cell trajectories, and the model parameters can be device parameters such as device length, width or height, and cell properties.

CHAPTER 1
INTRODUCTION AND BACKGROUND

System-level approach to describe and interpret interactions among parts of systems is very useful for understanding the mechanism of the system. This system-level approach provides benefits of controlling the behavior of the process of the system and simulating the prediction of the process. In addition, system-level approach allows us to find the emergent properties of the process, which are the results of interactions among the parts of the process [1, 2, 3]. These emergent properties cannot be found from a single part of the process, but rather from the entire process. These advantages have made the system-level approach applied in the biology field, which is called systems biology.

Systems biology is the system-level approach for describing and understanding the mechanisms of biological processes, such as interactions among proteins or genes [1, 2]. In systems biology, several mathematical forms have been used such as ODE-based model, stochastic model, Boolean network, and Bayesian network [4, 5, 6, 7, 8, 9, 10]. Each mathematical form can be selected according to the properties of the biological process. Since most biological processes include dynamic behaviors and the ODE-based model can describe the changes of system components over time, the ODE-based modeling is usually used. The ODE-based models are often constructed by including prior knowledge of interactions among individual genes and proteins in the complex system. The ODE-based model consists of several unknown model parameters such as reaction rates, binding affinity, Hill coefficient, etc [11, 12, 13, 14, 15], thus parameter estimation is an important step toward deeper understanding of the systems. In general, these unknown model parameters can be estimated based on the experimentally observed data.

When the ODE-based model and the experimental data are given, parameter estimation aims to find the optimal parameters which enable the model to reproduce the experimental

data. Parameter estimation is often formulated as a least squares optimization problem (Eq.1.1), where all experimental data points are typically considered as equally important. In Eq. 1.1, n represents the total number of experimentally observed data points, obs_i represents the value of the i^{th} observed data point, $pred_i$ represents the model prediction of the i^{th} data point which is a function of the parameters, θ.

$$Cost = \frac{1}{2} \sum_{i=1}^{n} (obs_i - pred_i(\theta))^2 \tag{1.1}$$

This cost function measures the differences between the experimental data and the simulated model prediction generated from the estimated parameters, and this difference is called the cost value. By minimizing the cost function value, we can find model parameters that enable the model to produce simulated data that are similar to the experimentally observed data.

Compared to the model complexity, the amount of available experiment data is usually insufficient to constrain several unknown model parameters. In this situation, it is possible that very different parameter settings can fit the data equally well [16]. This is a manifestation of an information gap between the model complexity and the data. This imbalance between insufficient available data and highly complex model makes the parameter estimation more challenging.

To overcome this information gap between the limited available data and the high model complexity, two strategies have been developed: Experimental design method and model reduction method. The goal of the experimental design method is to obtain more data by performing extra experiments [17, 18, 19]. More specifically, the experimental design method tries to identify the most informative experiment in improving parameter estimation, given the ODE-based model and existing experimental data. To find the most informative experiment among several candidate experiments, information theory, sensitivity analysis, and Bayesian posterior sampling have been used to define criteria for measuring

information contained in candidate experiments [17, 20, 21, 22, 23, 24]. The purpose of the model reduction method is to simplify the complex mode while maintaining its ability to fit the existing data. This method can be used when additional experiments cannot be performed because of some practical reasons such as high costs of running more experiments. Given the ODE-based model and experimental data, the model reduction method tries to find the way to reduce the complexity of the model while retaining the ability to fit the data. By exploiting special properties and sensitivities of the model, the part of the model, which can be simplified, is identified [25, 26, 27, 28, 29, 30, 31, 32, 33, 34]. Furthermore, by reducing the model, the key controlling part of the system can be found.

To sum up, the experimental design method identifies the most informative experiment among several candidate experiments. By conducting the most informative experiment, we can effectively reduce the information gap. On the other hand, the model reduction method simplifies the complex biological model by removing or combining the insensitive parameters without losing the ability to fit the given data. Moreover, by simplifying the model, the key mechanisms behind the systems behavior also can be identified. Previously, these approaches are considered very separately, however, these two methods share deep connections that can be unified into a common framework. Therefore, in Chapter 2, the possibility of unified framework for these two distinct methods will be discussed by using geometric concept. This unified framework has potential to reduce the information gap between limited data and complex mathematical model efficiently by identifying the most informative experiment and finding the key controlling mechanism of the system at the same time. The unified approach will lead to a new technique for bridging the information gap between limited experimental data and complex mathematical model in systems biology. In addition, we expect that this unified approach can bring intuitive geometric interpretation in systems biology.

In Chapter 3, a new approach to estimate more accurate model parameters will be introduced, when the experimental data is not enough to constrain the model. Based on what we

learned in experimental design method, we know that there are some good experiments that provide a lot of information, and there are less informative experiments that do not provide additional information. In the same manner, if we look at the data points collected in one single experiments, it is also possible that some data points are informative, and some data points are less informative. This is the conceptual motivation that the relative importance of each data point should be considered when estimating parameters.

The relative importance of each observed data point is quantified using the inverse fisher information matrix. If one data point can be estimated based on the other data points, then it means that this data point does not contain new information. Thus, this data point should be received a low weight. On the other hand, if we can predict one data point using the other data points, this data point contains new information that does not exist in other data points, so it should be given a large weight. To embed this relative importance of each data point, we re-formulate the least square cost function (Eq. 1.1). The benefit of this new approach is that performing additional experiments or verifying the reduced model is not required to improve parameter estimation. Previously, we needed to perform additional experiments selected by experimental design method. Also, we needed to reduce model complexity by combining or eliminating insensitive parameters to obtain more accurate parameter estimation results. However, when additional experiments are not available or the reduced model with retaining the ability to fit the observed data cannot be found, our new approach can be used to improve parameter estimation results.

In previous chapters, we have discussed several benefits and limitations of mathematical modeling. Also, several approaches to address the limitations have been demonstrated. In Chapter 4, we will demonstrate how to build a mathematical model to describe a cell sorting procedure within the microfluidic device, as an application. The reason why we select the microfluidic device as an application is that microfluidics is a promising technology for biological inquiries at the single-cell level, such as single-cell gene expression for lineage analysis, microfluidic cell sorting, and signaling dynamics [35]. Also, the mi-

crofluidic cell sorting device help us to distinguish the cells based on their stiffness. The stiff cells move up and the soft cells move down along the path of the cells flow. This is a very important property because it can be used to detect the cancer cells, such as breast cancer and leukemic cancer cells, which are typically softer than the normal cells [36]. Since the microfluidic cell sorting device is an effective tool for studying a single cell analysis, mathematical approach would greatly help in developing an accurate analytical tool and also in understanding the mechanisms of the cell sorting result within the device. Since the microfluidic cell sorting device is an effective tool for studying a single cell analysis, mathematical approach would greatly help in developing an accurate analytical tool. In this work, the ridge induced circulation microfluidic cell sorting device will be discussed. This microfluidic device is decorated with periodic diagonal ridges and this microfluidic device is usually used to sort or isolate the cells depending on cells biophysical properties, such as cell stiffness or cell size [37, 38, 39, 40]. Developing mathematical model describing cell trajectories within this microfluidic device will allow us to optimize the device parameters (e.g. angle of the ridges, width and height of the device) to achieve accurate cell sorting results. Furthermore, before performing experiments, we can simulate different cell trajectories according to different cell properties by using developed mathematical model.

CHAPTER 2
UNIFIED FRAMEWORK: EXPERIMENTAL DESIGN AND MODEL REDUCTION METHODS

2.1 Introduction

Mathematical modeling is an important tool for understanding complex biological systems. It is very useful for making predictions about system behavior far away from its nominal operating conditions [4, 5]. These models can take the forms of ordinary differential equations [9], Boolean networks [41], stochastic differential equations [42], Petri net [10], Bayesian networks [8], and other frameworks [43, 6, 7]. Each form can be selected according to properties of the biological processes they intend to describe. In systems biology, a popular modeling tool for the dynamics of biological networks is ordinary differential equations (ODEs). ODEs are describing changes of abundance or concentration of interacting components over time. ODE-based systems biology models are often constructed by including prior knowledge of interactions among individual genes and proteins in the complex system, resulting in highly complex models with many unknown parameters, such as reaction rates, Hill coefficient, etc [11, 12, 13, 14, 15]. In general, these unknown model parameters can be estimated based on the experimentally observed data. However, compared to the complexity of the models, the amount of experimentally available data is almost always limited and not enough to constrain the parameters. As a result, it is possible that very different sets of parameters can fit the data equally well. This is a manifestation of an information gap between the model complexity and the data [16]. In this situation, parameter estimation is an ill-posed problem, and thus very challenging.

In order to bridge this gap between high model complexity and limited experimental data, one strategy is to develop better optimization algorithms for parameter estimation,

and investigate sensitivity and identifiability to evaluate which parameters are accurately estimated and which ones have large uncertainty. Literature along this line is voluminous, and most progress took place in the fields of statistics, systems and control theory, and machine learning [44, 45, 46, 47, 48]. An alternative strategy is to make the problem better conditioned by obtaining more data or simplifying the model. This can be done with methods that are referred to as experimental design [17, 18, 19] and model reduction methods [49, 50, 51, 25] in the literature. This thesis focuses on discussing the second strategy, the experimental design and the model reduction perspectives.

Experimental design method is an intuitive approach to bridge the information gap between complex models and limited experimental data, and this can be done by performing new experiments to obtain more data. To design a new experiment, one typically needs to decide what perturbation, activation or inhibition, is to be applied to which components such as genes and proteins in the system. Also, we need to decide which components should be measured at which time points. These choices together constitute the design of a new experiment. The experimental design question is that what additional experiment is expected to be maximally informative in improving parameter estimation and reducing uncertainty when we know the ODE model and data?

Model reduction method is another intuitive approach, which aims to simplify the complex model to an extent that is compatible to the available experimental data. This approach can be used when additional experimental data cannot be performed for some practical reasons, for example, biological samples are unavailable or experiments are too costly. Model reduction method can be applied to derive simpler models that can describe the data with fewer parameters. Furthermore, model reduction can lead to a minimal model that cannot be further reduced without losing its ability to explain the data. This minimal model may indicate the key controlling mechanisms of the biological system. One challenge is to identify the appropriate reduction among a huge number of possible ways to write down reduced models, such as removing or combining parameters or variables. The model re-

duction question is that when an ODE model and limited experimental data are given, how to systematically derive a sequence of reduced models, each with one fewer degree of freedom, such that the reduced models retain the ability to fit the data?

Both questions have been studied in the literature. For instance, experimental design methods have been developed based on information theory [21, 52], Bayesian posterior sampling [53, 20, 22, 54], and sensitivity analysis [23, 17, 55]. Model reduction methods have been applied to complex biological systems by exploiting system properties, such as time scales [26, 27, 28, 25, 29, 51], modularity [56, 31, 30, 57] and sensitivity [58, 59, 60, 33, 34, 14]. Although experimental design and model reduction methods have been considered as distinct methods, these two methods share deep connections. There exist methods that can tackle both questions, such as the profile likelihood [61, 62] and the model manifold analysis [24, 63, 64]. To illustrate in details the processes of experimental design and model reduction, a simple dynamical system is used as an example using the profile likelihood and model manifold methods [65].

2.2 Experimental design method

2.2.1 Existing experimental design methods

When an ODE-based model and limited experimental data are given, experimental design aims to identify the experiment that is expected to be maximally informative in improving parameter estimation. Different existing experimental design methods differ by how ”maximally informative” is defined. For example, methods based on Bayesian posterior sampling aim to identify experiments which optimizes the expected value of some objective functions associated to candidate experiments [53, 20, 61, 54]. Information theoretic methods use entropy and mutual information to quantify the additional amount of information contained in candidate experiments [21, 52]. Methods based on sensitivity analysis aim to find experiments which can maximally reduce the variance and uncertainty of the estimated parameters [23, 66, 17, 67, 24, 68].

The Approximate Bayesian Computation method based on Sequential Monte Carlo (ABC-SMC) is a powerful approach for parameter estimation and model selection [69]. The ABC-SMC algorithm can be applied to sample the parameter space from a posterior distribution defined based on the fit between parameter values and experimental data. As an output of this sampling process, an empirical distribution of the parameters is obtained, which describes how well the parameters are constrained by the experimental data. Compared to many parameter optimization algorithms that provide a point estimate, the empirical distribution contains richer information, which can be used for experimental design. In a Bayesian active learning method [54], this empirical distribution was used to compute, for each candidate experiment, the expected value of a loss function defined by the error of the model predictions, and the expected loss was the criterion for selecting the optimal experiment to perform next. In another Bayesian method [20], the empirical distribution of parameters was used to compute the expected variance of estimated parameters if the data is augmented by each candidate experiment, and the expected variance served as the criterion for experimental design. In the profile likelihood method [61], a variation of this empirical distribution was used to calculate the range of model predictions for each candidate experiment, and suggested that the optimal experiment should be the one with the widest spread of predictions.

Information theoretic experimental design approaches incorporate entropy and mutual information measures into the Bayesian methods. The typical criterion for optimal experiment is to maximize the mutual information between parameters and candidate experiments. More precisely, the KullbackLeibler divergence between the prior distribution of the parameters and the posterior distribution of parameters given data from candidate experiments [21, 52]. The prior distribution of the parameters encodes either the prior knowledge of the parameters if no experimental data is available, or the posterior distribution of parameters given available data from previously performed experiments, allowing iterative procedures between the computational analysis of experimental design and the biological

efforts of carrying out the experiments.

Sensitivity analysis examines the derivatives of model predictions with respect to the parameters, which form the Jacobian and the Fisher Information Matrix (FIM) [70]. For each candidate experiment, a separate FIM can be constructed by considering the derivatives of model predictions associated to the available experiments and the candidate experiment. When the FIM is evaluated at a point estimate of the parameters given the currently available data, the resulting matrix represents a linear approximation of the model, and the properties of the FIM provide quantification of parameter uncertainty if the candidate experiment is performed, which can be used as criteria for experimental design [71]. One of these criteria is A-optimality, which minimizes the trace of the inverse of the FIM, and hence minimizes the variance of the estimated parameters [67, 24]. Another popular criterion is the D-optimality, which maximizes the determinant of the FIM [23, 66, 17, 68]. Although the linear approximation seems insufficient in handling complex nonlinear dynamical models, it has been shown to be an effective method for experimental design in many systems biology studies, and is computationally much more efficient compared to Bayesian approaches.

2.2.2 An example model: sum of two exponentials

A simple toy exmample model is introduced in this section to discuss experimental design in details [65]. We assume that we have a dynamical system that contains two exponentially decaying variables with unknown decay rates. Assume the two variables cannot be measured separately, and we can measure the sum of them at time points of our choosing. For simplicity, we further assume that at $t = 0$, the initial values of both variables are 1.

This system can be modeled by the following ODEs in equation (2.1).

$$\begin{cases} \frac{dx_1}{dt} &= -\theta_1 x_1 \\ \frac{dx_2}{dt} &= -\theta_2 x_2 \\ x_3(t) &= x_1(t) + x_2(t) \\ x_1(t=0) &= 1 \\ x_2(t=0) &= 1 \end{cases} \tag{2.1}$$

where x_1 and x_2 represent the two dynamical variables, which exponentially decay at rates θ_1 and θ_2, respectively. x_3 represents the sum of two variables and we can measure its value. Since this ODE model is simple and linear, its analytical solution in equation (2.2) can be calculated. However, for more complex systems biology models, such an explicit analytical solution is typically unavailable.

$$\begin{cases} x_1(t) &= e^{-\theta_1 t} \\ x_2(t) &= e^{-\theta_2 t} \\ x_3(t) &= e^{-\theta_1 t} + e^{-\theta_2 t} \end{cases} \tag{2.2}$$

We design an initial experiment, where we measure the sum x_3 at time points t=1 and t=3, to estimate the decay rates. Eq. 2.3 represents the mathematical description of the experiment.

$$\begin{cases} obs_1 &= x_3(t=1) \\ obs_2 &= x_3(t=3) \end{cases} \tag{2.3}$$

After we carry out this experiment, the observed measurements are $1.10 * 10^{-5}$ and $1.04 * 10^{-15}$. The corresponding to noise-free simulation using true parameter values are $\theta_1 = 14$, $\theta_2 = 11.5$.

Using this simple example model, we introduce a few terminologies: parameter space, and data space. Since the system has two unknown model parameters θ_1 and θ_2, the "parameter space" is 2-dimensional. A 45 degree line is drawn in the parameter space because

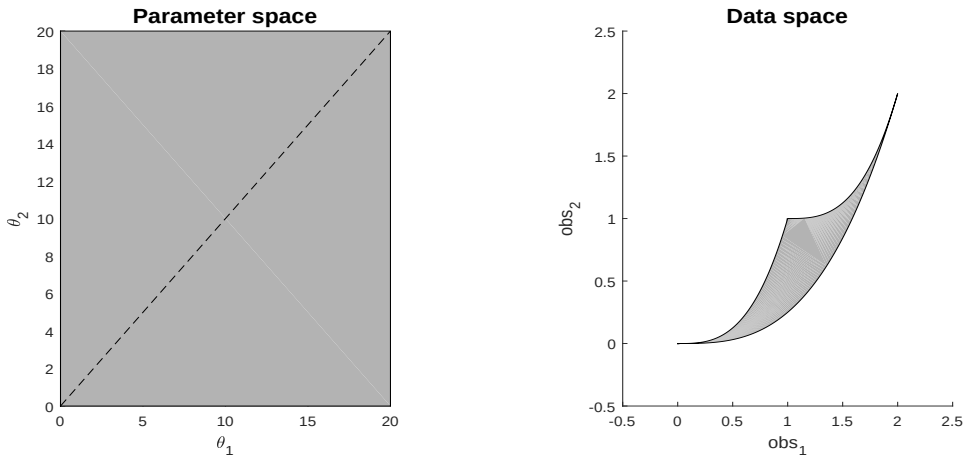


Figure 2.1: **The parameter space and the model manifold in the data space for the sum-of-two-exponentials example**

of the symmetry in the model. The experiment makes two measurements obs_1 and obs_2, and therefore the experimental data is a point living the a 2-dimensional "data space". The mathematical descriptions of both the system (2.1) and the data (2.3) together constitute the model, which defines a mapping from the parameter space to the data space. If we consider all parameter settings in the parameter space and map them to the data space, we will obtain a collection of points in the data space that are achievable by the model which is represented in Fig, 2.1 as a shaded area. We call these collected data points as a "model manifold". Typically, the model manifold does not occupy the entire data space, because of the constraints imposed by the mathematical structure of the model.

Albeit the simplicity of this example model, the fact of the experimental measurements being $1.10 * 10^{-5}$ and $1.04 * 10^{-15}$ represents a situation where the data does not contain enough information compared to the complexity of the model. There are two ways to explain this result. First of all, the measured variable x_3 is the sum of two exponential decays that will eventually go down to 0. By the time the two measurements are taken, x_3 is already extremely close to 0. Since the measurements are taken too late, there is little information in the observed data to infer the decay rates. Another one is that the observed data corresponds to one point in the data space, which sits in the bottom left corner of the

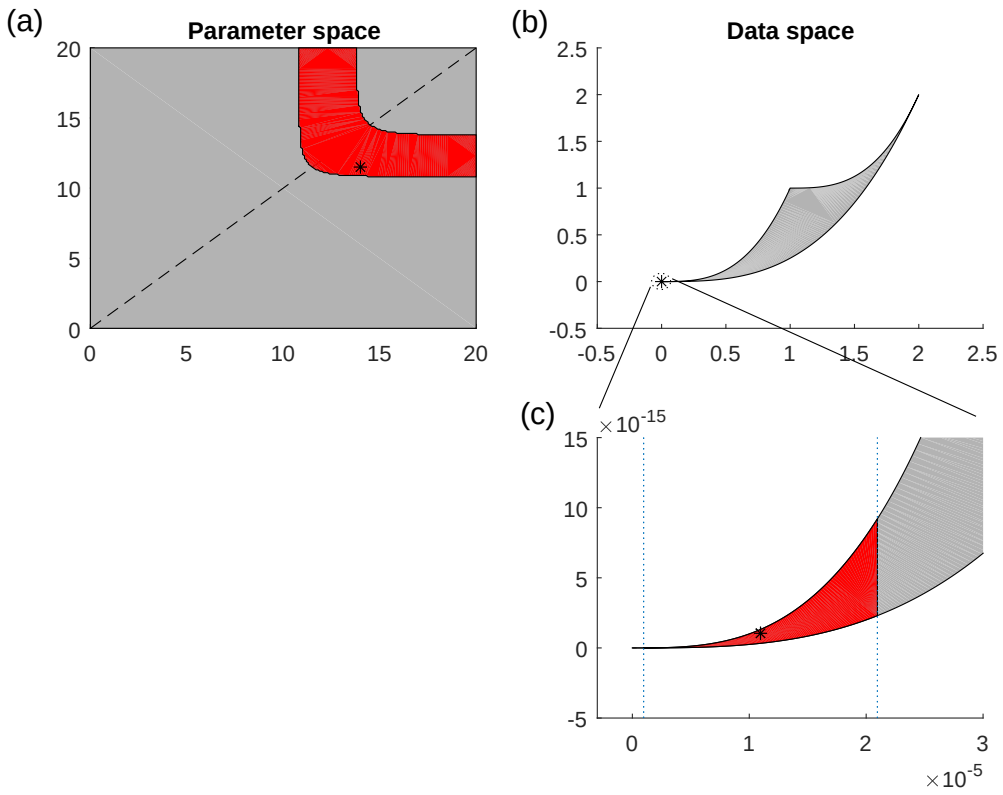


Figure 2.2: **Small region of the manifold can map to large region in the parameter space**

model manifold, as shown in Fig. 2.2b. If we formulate a least squares problem to estimate the parameters by an optimization algorithm such as gradient descent, interior point [44], and set a small error tolerance threshold of 10^{-5}, then the optimization algorithm will stop when it reaches a solution in the red region of the model manifold in Fig. 2.2b. In Fig. 2.2c, we can see that this region is tiny and only visible in the zoomed-in view. However, this tiny red region of the manifold corresponds to a large region in the parameter space shown in Fig. 2.2a. Therefore, the estimated parameter can land anywhere in the red region of the parameter space, far away from the true parameters indicated by the star, which means large uncertainty in the estimated parameters. In this example, parameter estimation is difficult because the available experiment is performed at an incorrect time scale. Intuitively, we should measure earlier time points. In the following, we use two experimental design methods to identify the appropriate time points we should measure.

2.2.3 Experimental design algorithm based on uncertainty reduction

An experimental algorithm based on sensitivity analysis and uncertainty quantification has been developed previously [24]. The main idea behind this algorithm is to identify the experiment that maximally reduces the uncertainty of the estimated parameter. The algorithm is as follows:

1. Set up a least squares cost function for parameter estimation $cost = \frac{1}{2} * \sum_i (obs_i - pred_i(\theta))^2$. Perform parameter estimation and obtain the best estimate based on currently available data.

2. Compute the Jacobian matrix $J_{i,j} = \partial pred_i(\theta)/\partial\theta_j$ at the best estimate. Compute the Fisher Information Matrix $I = J^T J$, which is an approximation of the Hessian of the cost function at the current best estimate. Compute the parameter uncertainty: $D = trace(I^{-1})$.

3. For each candidate experiment, compute the extended Jacobian, which is the Jacobian in step 2 appended by the partial derivatives of the model predictions of the candidate experiment with respect to the parameters. Compute parameter uncertainty based on the extended Jacobian. Suggest the experiment with smallest uncertainty as the next new experiment.

4. Carry out the suggested experiment to obtain new data.

5. Iterate through steps 1-4 to suggest new experiments until uncertainty cannot be reduced.

The data from the initial experiment are $x_3(t = 1) = 1.10 * 10^{-5}$ and $x_3(t = 3) = 1.04 * 10^{-15}$. The least squares cost function is formulated in step 1, and we optimize it using the Levenberg-Marquardt algorithm implemented in Matlab. The estimated model parameter values are $[10.70, 10.70]$. Although the initial experimental data corresponds to noise-free

simulation of the true parameter, the optimization algorithm stops early before reaching the true parameter due to numerical imprecision. However, the optimization achieves a very small value of $1.16 * 10^{-9}$ for the least squares cost. This disconnection between low cost value and incorrect model parameters is a manifestation of the fact that the data is limited compared to the complexity of the model. In step 2, the Jacobian matrix is computed which is composed of derivatives of experimental observations with respect to the parameters, evaluated at the estimated parameter. In this example, these derivatives can be written analytically in equation (2.4).

$$J = \begin{bmatrix} \frac{\partial x_3(t=1)}{\partial \theta_1} & \frac{\partial x_3(t=1)}{\partial \theta_2} \\ \frac{\partial x_3(t=3)}{\partial \theta_1} & \frac{\partial x_3(t=3)}{\partial \theta_2} \end{bmatrix} = \begin{bmatrix} -e^{-\theta_1} & -e^{-\theta_2} \\ -3e^{-3\theta_1} & -3e^{-3\theta_2} \end{bmatrix} \tag{2.4}$$

After evaluating the Jacobian at the current estimated parameter $[10.70, 10.70]$ and computing the Fisher Information Matrix and its trace, we can see that the uncertainty of the estimated parameter is very large, $4.93 * 10^{44}$. In complex mode where the analytical solution of the Jacobian is not available, the derivatives in the Jacobian can be approximated by finite differences or the sensitivity equations [72]. Although we notice the huge uncertainty associated with the estimated parameter, we do not have evidence against the estimated parameter because of the low cost value. We do not have reasons to suggest the estimated parameter is incorrect because we just do not have much confidence in it. Therefore, we assume the estimated parameter is correct, and design a new experiment that maximally increases our confidence by reducing the uncertainty. When the new experiment is carried out, we may realize that our assumption is wrong, and we can perform parameter estimation again based on the current and additional data to update the estimated parameter. In this example, the collection of all possible new experiments is defined as measuring x_3 at another time point. For each possible new experiment, at each time point, we compute the Jacobian associated to the existing experiments and the new experiment. Then we evaluate it at the current estimated parameter and compute the trace of the Fisher Information Matrix

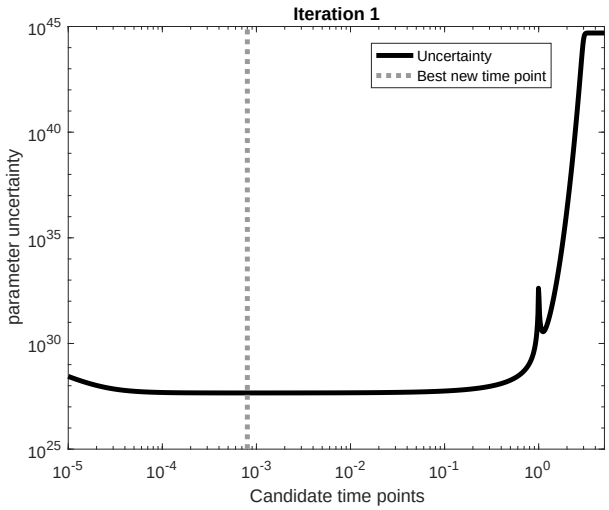


Figure 2.3: **Uncertainty at each candidate time point** In the first experimental design iteration, parameter uncertainty at each candidate time point. The time point $t = 0.0008$ is chosen for the next experiment because it leads to lowest uncertainty.

to quantify the parameter uncertainty if the new experiment is performed.

Fig. 2.3 shows the resulting uncertainty at each candidate time point. The peak at $t = 1$ shows that repeating a previous measurement does not efficiently reduce the uncertainty. The flat region after $t = 3$ indicates that measuring any time points after $t = 3$ does not reduce the uncertainty at all. Because $t = 3$ is already too late with observed $x_3(t = 3)$ extremely close to 0, which does not provide any additional information about the underlying parameter. The lowest uncertainty is achieved at $t = 0.0008$, which is the new experiment that our algorithm suggests. After the suggested experiment is performed, the mathematical description of the experiment becomes $obs_1 = x_3(t = 1)$, $obs_2 = x_3(t = 3)$, and $obs_3 = x_3(t = 0.0008)$. The experimental data becomes $[1.10 * 10^{-5}, 1.04 * 10^{-15}, 1.9797]$. Parameter estimation and uncertainty quantification in steps 1 and 2 are repeated. After that estimated parameter becomes $[3.85, 11.65]$, which is closer to the true parameter and has smaller uncertainty $3.30 * 10^{10}$.

Repeating step 3 leads to the design of the next experiment. In Fig. 2.4a, we can see that the uncertainty landscape changes, very early time points are no longer very useful because we already have a measurement taken at $t = 0.0008$. The uncertainty after t=1 is

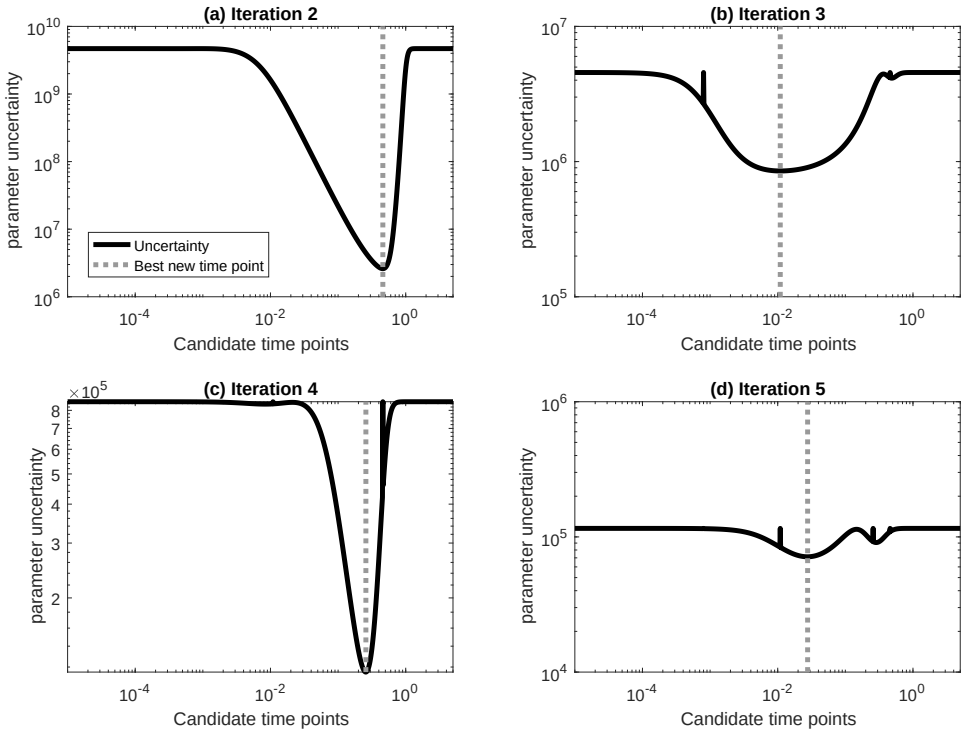


Figure 2.4: **Parameter uncertainty at each candidate time point in iterations 2∼5** The selected time points are 0.4589, 0.011309, 0.2576 and 0.0277, respectively.

flat and it is indicating that a new time point between 1 and 3 is not useful any more. The maximal uncertainty reduction occurs at $t = 0.46$, which is in a gap between the available time points. After the suggested experiment is performed, the entire experimental design process can iterate to suggest new time points as shown in Fig. 2.4b-d.

Fig. 2.5 shows five iterations of the experimental design algorithm based on uncertainty reduction. Fig. 2.5a shows the suggested time point at each iteration. Fig.2.5b shows that the estimated parameter is almost identical to the true parameter after the second experimental design iteration. In Fig. 2.5c, after the second iteration is completed, subsequent iterations only lead to moderate amounts of reduction in parameter uncertainty. This result is suggesting that the experimental data after the second iteration becomes sufficient compared to the complexity of the model, and the subsequent iterations are unnecessary.

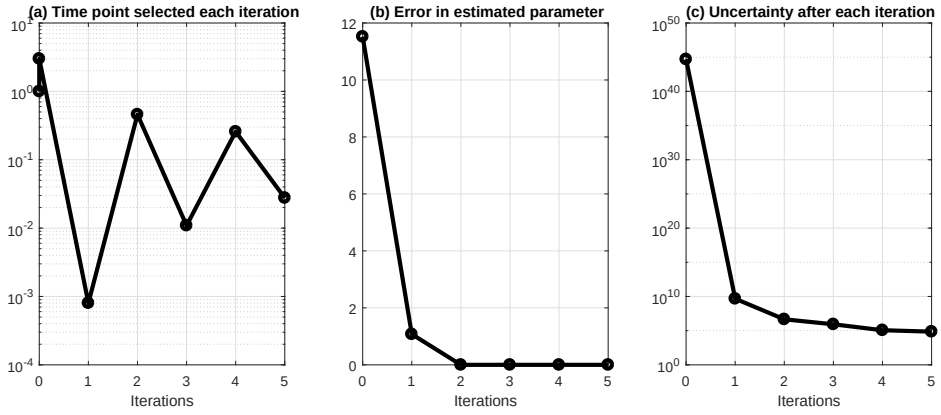


Figure 2.5: **Summary of five iterations of experimental design based on uncertainty reduction**

2.2.4 Experimental design algorithm based on profile likelihood

Experimental design of the previous example model can be performed using another algorithm based on the profile likelihood. In [61], the likelihood is defined with a Gaussian assumption,

$$Likelihood(obs|\theta) = \prod_i \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(obs_i - pred_i(\theta))^2}{2\sigma_i^2}} \tag{2.5}$$

where σ_i represents measurement noise. If we assume all measurements are equally accurate, the negative loglikelihood can be simplified as the following:

$$NegativeLogLikelihood(obs|\theta) = \frac{1}{2}\sum_i (obs_i - pred_i(\theta))^2 \tag{2.6}$$

which is the same as the cost function used in the previous experimental design method. The likelihood function in Eq. 2.5 or the negative loglikelihood cost function in Eq. 2.6 can be used to define the profile likelihood. The profile likelihood for each parameter is a function of the model parameter, defined by solving many optimization problems. We fix θ_1 to be a certain value, then we optimize the negative log-likelihood cost function with respect to the remaining model parameters, $\theta_2, ..., \theta_d$, where the d represents the number of unknown parameters. This optimized cost function identifies the best cost function value

and the estimated values for the remaining parameters. If we vary the value of θ_1 across its feasible range and perform the optimization for each possible value of θ_1, two curves are obtained. First one is the optimal cost as a scalar function of θ_1, and the other is the optimal values for the remaining parameters as a vector function of θ_1. The frist curve is the profile likelihood of θ_1 and it characterizes the optimal cost function values when θ_1 is fixed. The same analysis can be performed for each model parameter. As a result, we obtain d profile likelihoods, where d is the number of parameters.

For each profile likelihood, we can find the minimum value. Also, we can define a confidence interval around the best value of θ_1. For example, a generous definition would be a threshold at the 90 percentile of the optimized cost function values, which exclude 10 percent of possible θ_1 values with optimized cost function larger than the threshold. The accepted θ_1 values and the corresponding estimated values for the remaining parameters form a collection of acceptable parameter settings derived from the profile likelihood of θ_1. If we perform the same analysis for the profile likelihood of all the model parameters, we will obtain a larger collection of acceptable parameter settings. These acceptable parameter settings can then be used for experimental design. For one candidate experiment, we can use these acceptable parameter settings to simulate the experiment, and examine the range of the simulated data. If the range is small, all these acceptable parameters lead to similar predictions of the data that will be generated by the candidate experiment, which means this candidate experiment does not have the ability to differentiate the acceptable parameters. On the other hand, if the range is large, this candidate experiment is able to differentiate acceptable parameters and further constrain the parameters. Here is an algorithm for performing experimental design based on the profile likelihood:

1. Compute profile likelihood for each parameter. Basically, pick one parameter and vary its value in the feasible region. For each feasible value of the picked parameter, perform optimization to estimate the other parameters. The optimization objective is a least squares cost function $c = 0.5 * \sum_i (obs_i - pred_i(\theta))^2$.

2. Define thresholds to obtain the collection of acceptable parameter settings from the profile likelihood of each parameter.

3. For each candidate experiment, simulate the model prediction using all the acceptable parameter settings, and compute the range of the simulated predictions. Suggest the experiment with the largest range as the next new experiment.

4. Carry out the suggested experiment to obtain new data.

5. Iterate through steps 1-4 to suggest new experiments until some stopping criterion (the profiles become very sharp, or the range of simulated predictions becomes very tiny).

In this example model, we formulate the negative loglikelihood cost function in step 1 and compute the profile likelihood for each parameter using the Levenberg-Marquardt optimization algorithm implemented in Matlab. First, we fix θ_1 at different values, optimize the cost function with respect to θ_2, and generate the profile likelihood for θ_1 shown in Fig. 2.6a. In Fig. 2.6a, when θ_1 is fixed at small values, the optimized cost function is large because it is impossible to tune θ_2 alone to achieve small cost function value. Similarly, in Fig. 2.6c, the optimized θ_2 exhibits a strange oscillating pattern when θ_1 is fixed at small values, which is likely caused by numerical issues of the optimization algorithm. Since the two parameters are symmetric, the profile likelihood of the two parameters are identical, and therefore Fig. 2.6b and Fig. 2.6d are identical to Fig. 2.6a and Fig.2.6c.

To define acceptable parameter settings in step 2, a threshold should be selected. In this example model, the threshold is defined as 90% of the best likelihood in Eq. 2.5. The corresponding threshold on the profile likelihood is shown by the horizontal line in Fig. 2.6a and Fig.2.6b. This is a generous threshold because it is accepting many parameter settings along the profile likelihood. The acceptable parameter settings are shown in black in Fig. 2.6c and Fig. 2.6d, whereas gray indicates the parameter settings that are not accepted according to the threshold. In step 3, the model is simulated using the acceptable

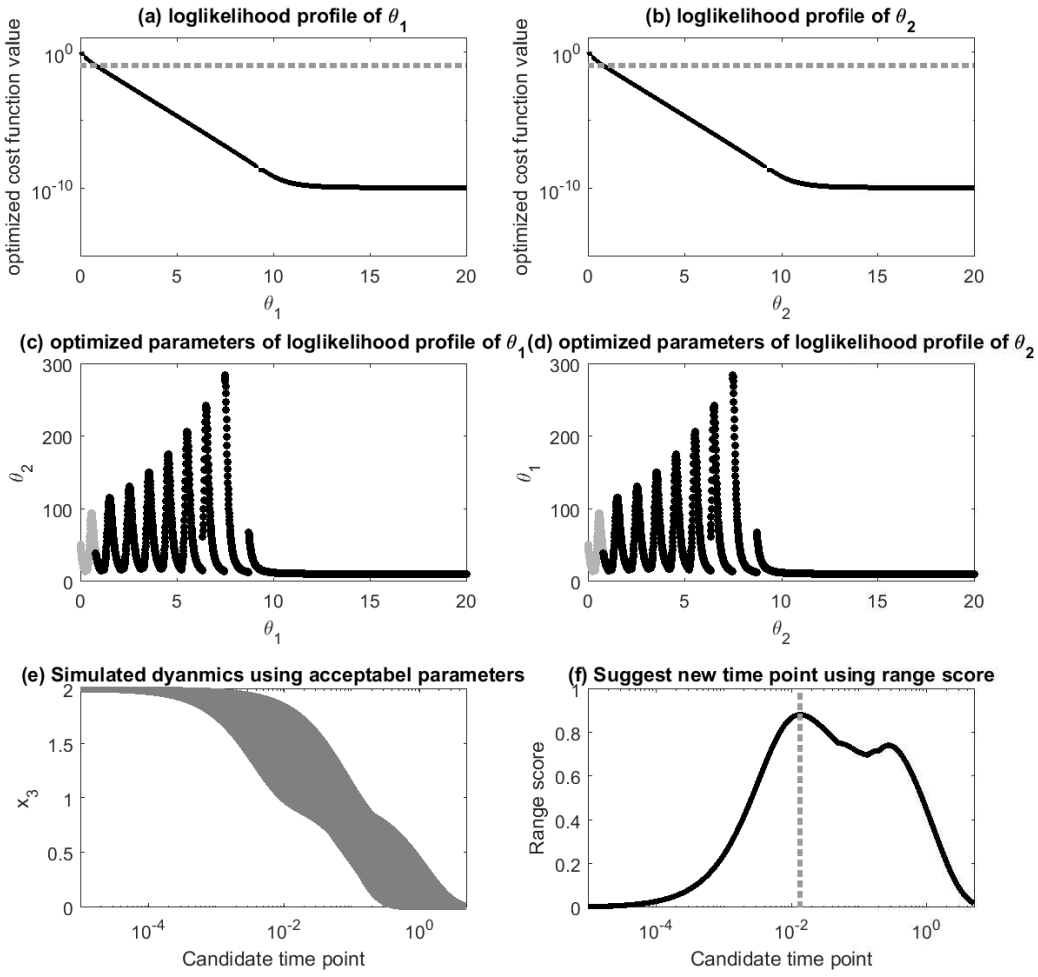


Figure 2.6: **The first iteration of profile likelihood based experimental design**

parameter settings to obtain the range of predicted data for all candidate experiments. The Fig. 2.6e shows the range of x_3's behaviors across the acceptable parameter settings. The time point $t = 0.0135$ is suggested as the new experiment, because it corresponds to the largest range, and hence has the most discriminating power among the acceptable parameter settings.

After performing the suggested experiment, the experimental data becomes $[obs_1, obs_2, obs_3] = [x_3(t = 1), x_3(t = 3), x_3(t = 0.0135)] = [1.10 * 10^{-5}, 1.04 * 10^{-15}, 1.6851]$. If we perform parameter estimation again with the new experimental data, the estimated parameter be-

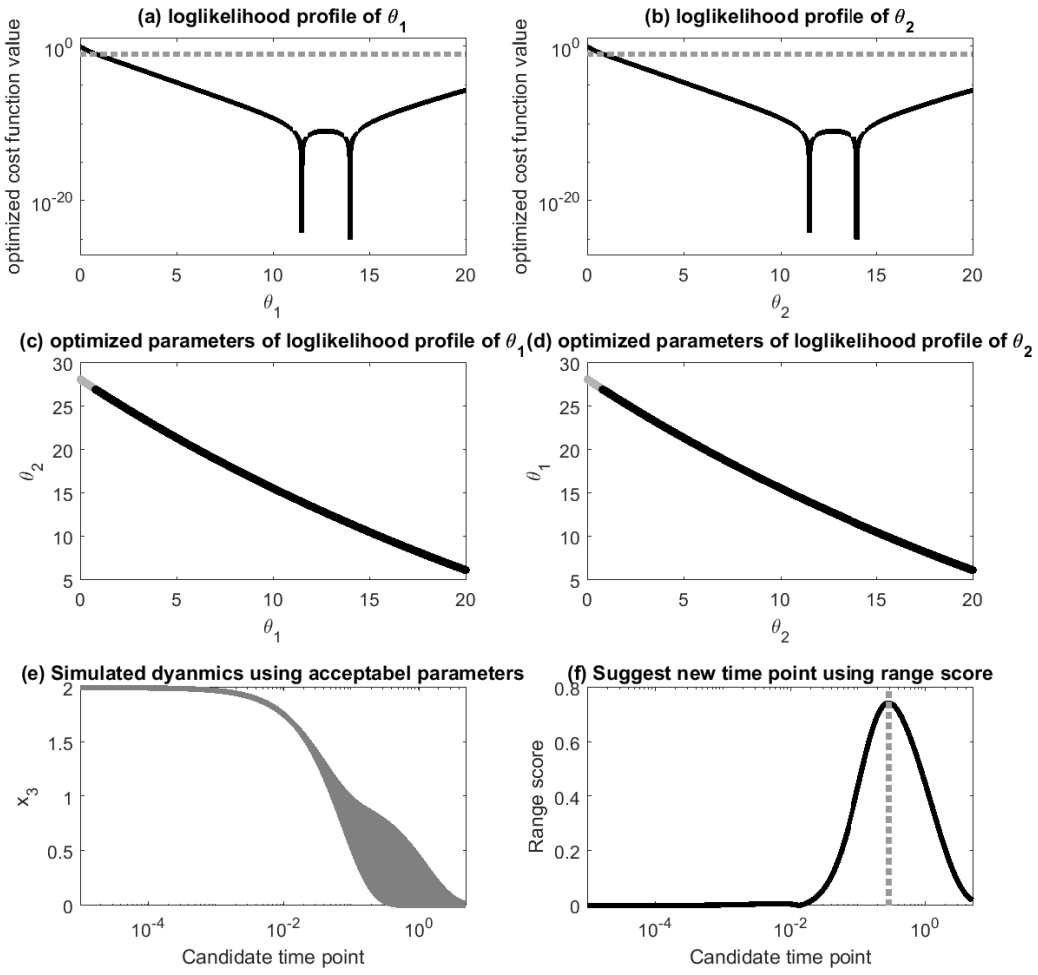


Figure 2.7: **The second iteration of profile likelihood based experimental design**

comes $[14.3431, 11.1697]$, closer to the ground truth parameters. We can perform a second iteration of the experimental design algorithm to suggest the next time point. In the second iteration, the updated profile likelihood is shown in Fig. 2.7a and Fig.2.7c. In these figures, the sharp peaks indicate that the current data is already able to constrain the parameters close to the true parameter values. In Fig. 2.7e, we can see that the simulated behaviors of x_3 still exhibit large range at late time points, and the algorithm suggests $t = 0.2899$ as the next time point to measure. A summary of the first five iterations of the profile likelihood based experimental design is shown in Fig. 2.8. The Fig. 2.8a shows the suggested time

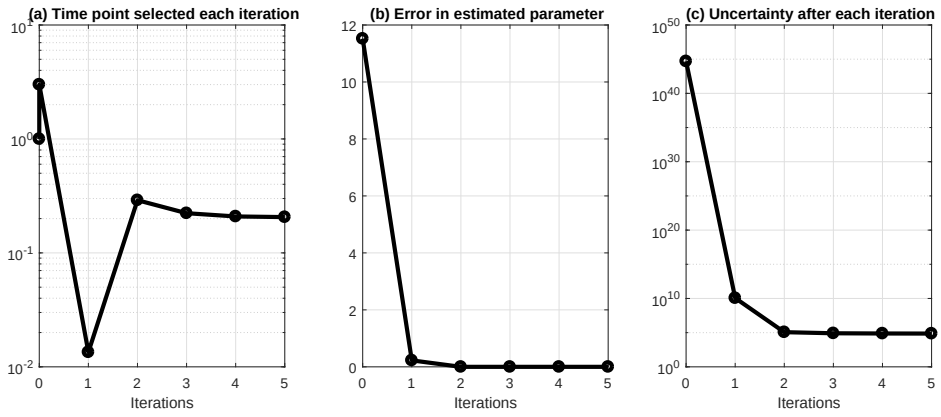


Figure 2.8: **Summary of five iterations of experimental design based on profile likelihood**

points in each iteration. After the second iteration, the algorithm stops exploring the new time points. This coincides with Fig. 2.8b, showing that the error in the estimated parameters approaches 0 after the second iteration. Although profile likelihood does not consider parameter uncertainty, the suggested time points do lead to great reduction of parameter uncertainty along the process, as shown in Fig. 2.8c.

2.2.5 Geometric interpretation of experimental design

In each iteration of the experimental design process, a new experiment is suggested and performed. More available data means that the dimension of the data space is increased. Since the number of parameters stays the same, the intrinsic dimension of the model manifold does not change. Therefore, increasing the data by new experiments will expand the data space by adding new dimensions, and deform the model manifold into the new dimensions, but will not change the dimension of the model manifold itself.

Fig. 2.9,2.10, and 2.11 illustrates how experimental design changes the geometry of the model manifold using the sum-of-two-exponentials example. With the initial experimental design $obs = [x_3(t = 1), x_3(t = 3)]$, the model manifold is a 2D object that lives in the 2D data space, shown in Fig. 2.9. The model manifold is colored by the sum of the two parameters. The upper right corner which is represented by blue color corresponds to both

parameters being small. The bottom left corner which is indicated by red color corresponds to both parameters being large. The data of the initial experiment sits in the bottom left corner of the manifold. After the first iteration of experimental design based on uncertainty reduction, a third observation $x_3(t = 0.0008)$ is made, and the model manifold becomes a 2D object in the 3D data space shown in Fig. 2.10, colored in the same way. If we look at the model manifold in Fig. 2.10 from top to down, we will see the exact same figure as the 2D version in Fig. 2.9. From the color, we can see that the new experiment expanded the bottom left corner (red region) of the model manifold and stretched that region downwards, while the rest of the model manifold (blue region) is less affected. This is because the new experiment is designed to improve parameter estimation of the data that sits at the bottom left corner. Fig. 2.11 shows the geometry of the model manifold after the first iteration of the experimental design algorithm based on profile likelihood, which is qualitatively the same as Fig. 2.10. As the experimental design algorithm iterates, the model manifold is expanded to higher dimension. The manifold region around where the experimental data sits is expanded more than the rest of the manifold.

When applied to the example model, the above experimental design algorithms show similar performance. In both cases, the first two suggested time points are sufficient in constraining the parameters, and therefore, the subsequent iterations are actually unnecessary. The two algorithms are comparable in terms of the error and the uncertainty in the estimated parameters. In this example, we assume the collection of all possible new experiments is defined by measuring x_3 at any time point. This is certainly unrealistic, because getting data very frequently at highly precise time points is very difficult experimentally. In reality, the collection of all possible and realistic new experiments is typically a smaller set due to experimental and technical constraints, and the uncertainty quantification for experimental design should only consider the realistic experiments.

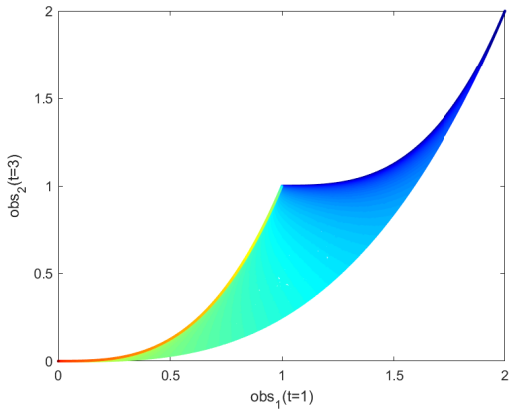


Figure 2.9: **Geometry of model manifold is changed by the experimental design process** Initial experimental design

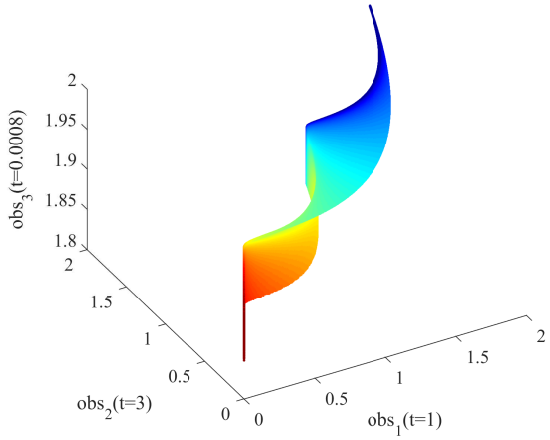


Figure 2.10: **Geometry of model manifold is changed by the experimental design process** Third time point at 0.0008

2.3 Model reduction method

2.3.1 Existing model reduction methods

Model reduction method aims to derive reduced models that can fit the data with fewer parameters. Furthermore, it also aims to identify the minimum model to elucidate the key mechanisms that give rise to the experimental observations. One challenge is to identify the appropriate reduction among a huge number of possible ways to write down reduced models such as removing or combining parameters or variables. Most existing model re-

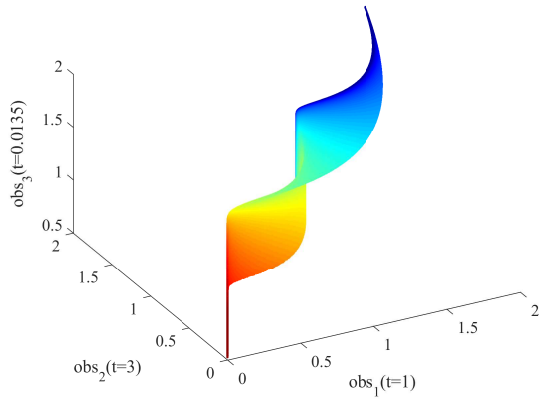


Figure 2.11: **Geometry of model manifold is changed by the experimental design process** Third time point at 0.0135

duction methods follow a two-step process. First, they identify which part of the model should be simplified by exploiting special properties and sensitivities of the model, and then we can write down the mathematical form of the simplified model using biological insights. Useful properties for model reduction include separation of time scales [73, 28, 26, 27, 25, 29, 51], clustering and lumping of variables into modules [56, 31, 30, 57], and insensitive parameters or variables [58, 59, 60, 33, 34, 14, 74].

Biological systems often contain processes that occur in different time scales [75]. For example, processes that occur much faster than the experimentally observed behavior can be assumed to be in the steady state. On the other hand, processes that occur much slower than observed behavior can be approximated as constants [76]. In an analysis of the Wnt/β-Catenin signaling pathway [26], the experimental observations are made on a time scale of hours. Since separate experiments observed no detectable degradation of several proteins in the pathway in several hours, the concentration of these proteins are assumed to be constants, so that the corresponding differential equations are converted into algebraic equations, reducing the number of dynamical variables. In addition, a few reversible binding reactions are known to occur much faster, which lead to quasi-equilibrium approximations that turn reaction rates into ratios, reducing the number of parameters [26]. For complex models, computational singular perturbation analysis can be applied to systemat-

ically identify processes or transformations associated to fast and slow time scales, which lead to model reduction [73, 28, 27, 29, 51].

Lumping methods for model reduction aims to remove dynamical variables from the system and replace them by new lumped dynamical variables that represent affine combinations of the removed variables [56, 31]. Variables to be lumped can be identified by either biological intuition of the structural properties of the model [77, 78, 79, 30, 50], or systematic algorithms based on principle component analysis [80, 81], greedy iterative forward selection [57] and decomposition algorithms [13]. One special case of lumping methods is called proper lumping, where each variable in the system contributes to only one of the new lumped variables. This constraint allows the interpretation of the reduced model to be clearly connected to the interpretation of the original variables [30, 57, 13].

Sensitivity analysis is another popular approach for model reduction. In complex models with limited experimentally observed data, there are typically parameters or parameter combinations that are not important for the observed data. In sensitivity analysis, the insensitive model parameters that affect the dynamics the least are eliminated [58, 59, 60]. The insensitive parameters can be identified by principal component analysis of the Jacobian [34, 14] and flux analysis of the stoichiometry [33].

2.3.2 The sum-of-two-exponentials example for model reduction

Similar to the discussion of experimental design method, we use the sum-of-two-exponentials model to illustrate model reduction methods. The model reduction problem starts with the mathematical model in Eq. 2.1) and Eq. 2.3. Similar as before, we assume the true underlying parameter values are $\theta_1 = 14, \theta_2 = 11.5$, and the experimental data are $obs_1 = x_3(t = 1) = 1.10 * 10^{-5}$ and $obs_2 = x_3(t = 3) = 1.04 * 10^{-15}$. Therefore, the model and data together represent a situation where the data is not enough to constrain the model parameters, making this problem amendable to model reduction. In the following, we apply two model reduction methods to derive reduced models for this example.

2.3.3 Model reduction based on Manifold Boundary Approximation Method (MBAM)

The Manifold Boundary Approximation Method (MBAM) [63] performs the model reduction by manifold boundaries. The main idea of this method is to view the experimental data as a point in the data space, project it on the model manifold, and identify the nearest manifold boundary. This manifold boundary corresponds to a reduced model with one fewer degree of freedom, and is able to better fit the data than other reduced models corresponding to other boundaries of the manifold.

In this example model, the model manifold has three boundaries, which are represented in Fig. 2.12. Starting from the original model in equations (2.1) and (2.2), if θ_1 is set to infinity, the solution of the model becomes $x_3(t) = e^{-\theta_2 t}$, as the exponential term corresponding to infinite θ_1 decays to 0 instantaneously. Given the same initial experiment that makes two measurement, the model manifold for the reduced model becomes a 1D object in the 2D data space, which is the blue boundary of the original model manifold in Fig. 2.12. Setting θ_1 is close to 0 leads to another reduced model $x_3(t) = e^{-\theta_2 t} + 1$, whose corresponding model manifold is the green boundary in Fig. 2.12. Lastly, if the two decay rates are equal $\theta_1 = \theta_2$, the original model reduces to $x_3(t) = 2e^{-\theta_2 t}$, and the model manifold reduces to the red boundary. This example shows the boundaries of the original model manifold correspond to reduced models that can be derived by physically meaningful limits of the parameters.

The MBAM algorithm [63] is as follows:

1. Based on the current model, set up a least squares cost function for parameter estimation $c = 0.5 * \sum_i (obs_i - pred_i(\theta))^2$. Perform parameter estimation and obtain the best estimated parameter based on the data, which corresponds to the projection of the experimental data onto the model manifold.

2. Identify the manifold boundary nearest to the experimental data by numerical integration of the geodesic equation, using the estimated parameter as the 0'th order

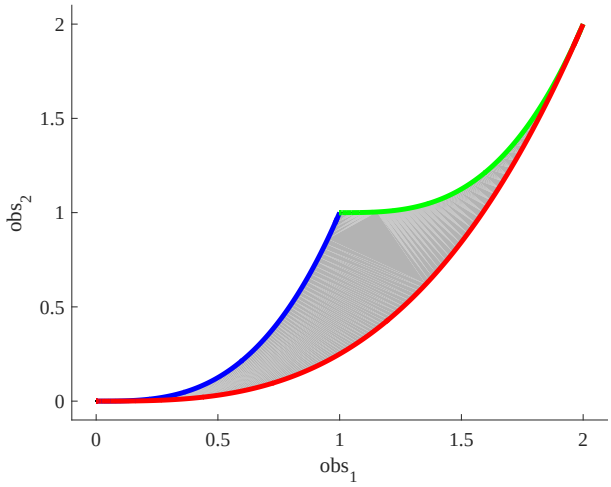


Figure 2.12: **Model manifold boundaries**

initial condition and the least sensitive parameter directions from the Fisher Information Matrix as the first order initial condition. The geodesic equation is a set of N nonlinear second-order differential equations: $\frac{d^2\theta^\mu}{d\tau^2} + \sum_{\alpha,\beta} \Gamma^\mu_{\alpha,\beta} \frac{d\theta^\alpha}{d\tau} \frac{d\theta^\beta}{d\tau} = 0$, where μ runs from 1 to N, indexing the parameters. $\Gamma^\mu_{\alpha,\beta}$ is known as the connection coefficient. $\Gamma^\mu_{\alpha,\beta} = \sum_{v,m}(g^{\mu v} \frac{\partial pred_m}{\partial \theta^v} \frac{\partial^2 pred_m}{\partial \theta^\alpha \partial \theta^\beta})$ where, $g^{\mu v} = (\sum_m \frac{\partial pred_m}{\partial \theta^\mu} \frac{\partial pred_m}{\partial \theta^v})^{-1}$ is the (μ, v) element of the inverse of the Fisher Information Matrix. In the summations, α, β, v run from 1 to the number of parameters, and m runs from 1 to the number of experimental observations. If we set $\theta(\tau = 0)$ to be the estimated parameter and set $\theta'(\tau = 0)$ to be the eigenvector of the smallest eigenvalue of the Fisher Information Matrix (the least sensitive parameter direction), solution to the geodesic equation is a nonlinear curve in the parameter space which maps to a "straight" line on the model manifold, leading to the boundary closest to where the projection of the experimental data sits.

3. As the geodesic path approaches a boundary, the Fisher Information Matrix becomes singular, and certain parameters approach to limiting values (such as 0, ∞). We evaluate the limits and manually write down the mathematical form of a reduced model.

4. With the reduced model, go through steps 1-3 to derive further model reductions, until the reduced model cannot fit the experiment data well.

One challenging step of MBAM is the numerical integration of the geodesic equation, which is a set of second-order differential equations. In orde to use existing differential equation solvers to perform numerical integration, the geodesic equation can be changed into the following first-order form.

$$\frac{d}{d\tau}(\theta^\mu) = \left(\frac{d\theta^\mu}{d\tau}\right), \mu = 1, 2, ..., N \tag{2.7}$$

$$\frac{d}{d\tau}\left(\frac{d\theta^\mu}{d\tau}\right) = -\sum_{\alpha,\beta} \Gamma^\mu_{\alpha,\beta} \frac{d\theta^\alpha}{d\tau}\frac{d\theta^\beta}{d\tau}, \mu = 1, 2, ..., N \tag{2.8}$$

Denote $\frac{d\theta^\mu}{d\tau}$ as new variables $[d\theta^\mu]$, equation (2.8) becomes clearer.

$$[\dot{\theta^\mu}] = [d\theta^\mu] \tag{2.9}$$

$$[\dot{d\theta^\mu}] = -\sum_{\alpha,\beta} \Gamma^\mu_{\alpha,\beta} [d\theta^\alpha][d\theta^\beta] \tag{2.10}$$

The dot above represents the first order derivative with respect to τ. Numerical integration of Eq.2.10 requires the connection coefficients $\Gamma^\mu_{\alpha,\beta}$, which are functions of the parameters θ. As mentioned in step 2 of MBAM, calculating the connection coefficients involves the first- and second-order partial derivatives of the model predictions of the experimental data with respect to the parameters, which can be computed by finite differences or the sensitivity equations [72]. Overall, the process of integrating the geodesic equation is the following: first specify an initial parameter and initial velocity, numerically compute the connection coefficients at the initial parameter, inch forward both θ^μ and $d\theta^\mu$ using Euler formula, then recompute the connection coefficient at the new θ^μ, inch forward again, and keep repeating this process to obtain the geodesic path in the parameter space.

Since MBAM uses manifold boundaries to derive reduced models, it requires the model manifold to be bounded and requires the model to be differentiable. In general, mathematical models are usually constructed with biological assumptions and constrained by bounded production, exponential decay or mass conservation. Since these ofthen make the model manifold bounded, MBAM is generally applicable to models describing biological processes. The manifold for the sum-of-two-exponentials example is obviously bounded, and hence can be analyzed using MBAM. In the sum-of-two-exponentials example, the decay parameters are non-negative. Many models in systems biology also involve non-negative constraints on the parameters. For such constraints, one neat trick is to modify the model to work with log-parameters, which are unconstrained because the exponentiation automatically takes care of the non-negative constraints. The modified model in log-parameters is shown in equation (2.11):

$$\begin{cases} x_3(t) &= e^{-te^{log\theta_1}} + e^{-te^{log\theta_2}} \\ obs_1 &= x_3(t=1) \\ obs_2 &= x_3(t=3) \end{cases} \tag{2.11}$$

where the two parameters are $log\theta_1$ and $log\theta_2$.

In this example model, we assume that the parameter estimation in step 1 is perfectly accurate. Also, the true parameter values are used as the initial parameter for integrating the geodesic equation, $log\theta^\mu = [log(14), log(11.5)]' = [2.6391, 2.4423]'$. The initial velocity $dlog\theta^\mu$ is the least sensitive parameter direction. To obtain the initial velocity, we compute the Jacobian matrix J at the initial parameter,and then we compute the Fisher Information Matrix $J'J$. After that we find its eigenvalues and eigenvectors. The eigenvector corresponding to the smallest eigenvalue is the initial velocity direction. Equivalently, the initial velocity direction can be defined by the right singular vector of J that corresponds to J's smallest singular value. The eigenvector (or the singular vector) is not the initial velocity yet, there is still ambiguity about the direction, because the opposite direction of an

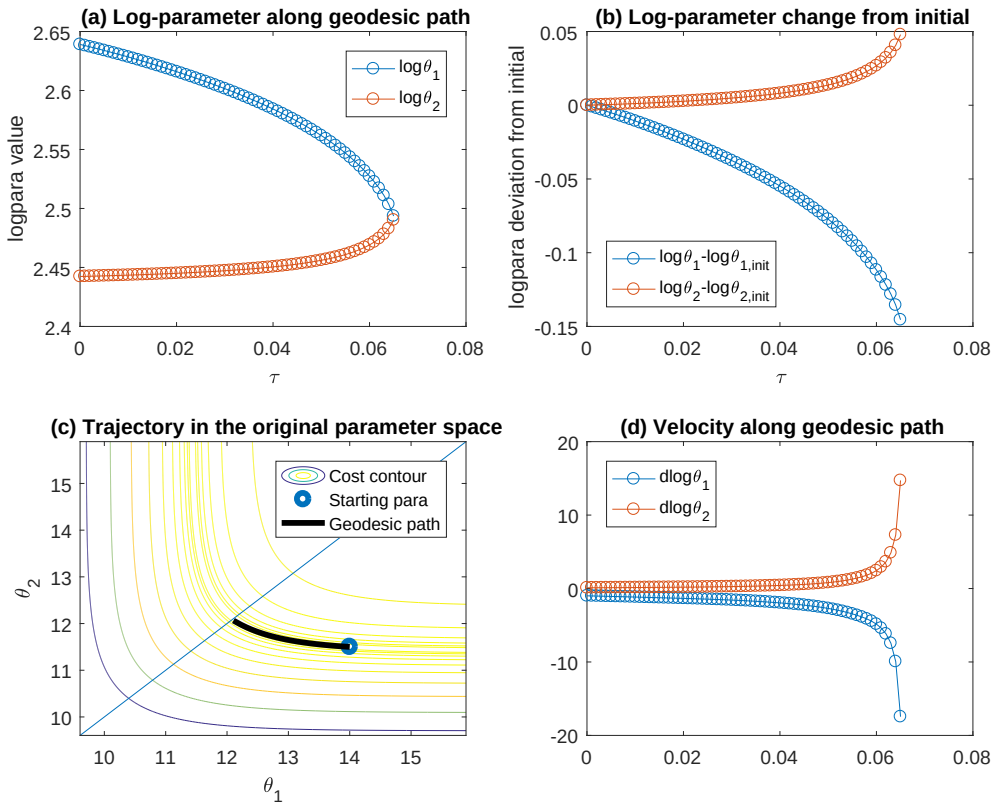


Figure 2.13: **Geodesic path obtained in the first iteration of MBAM model reduction for the sum-of-two-exponentials model** (a) The log-parameter values along the geodesic path. (b) The deviation of log-parameter from the initial point. (c) The geodesic path in the original parameter space overlaid with the cost surface contours. (d) The velocity of the log-parameters along the geodesic path.

eigenvector is also an eigenvector corresponding to the same eigenvalue. To determine the direction, we compute the right hand side of the $[d\dot{log}\theta^{\mu}]$ equation in (2.10), which defines the acceleration. If the inner product of the eigenvector and the acceleration is positive, we define the initial velocity to be the eigenvector itself. If the inner product is negative, we define the initial velocity to be the negative of the eigenvector. In this case, the initial velocity is $[-0.9950, 0.0994]$.

After determining the initial conditions of the geodesic Eq. 2.10, we numerically integrate it to identify the nearest boundary. Fig. 2.13a shows the trajectory obtained by integrating the geodesic equation, the log-parameter values as functions of τ which pa-

rameterizes the geodesic path. We can see that the two parameters gradually approach each other and become almost the same. In Fig. 2.13c, we show the geodesic path in the original parameter space, overlaid with the least squares cost surface. We can see that the geodesic path starts from the initial parameter we specified, and moves along the canyon of the cost surface until some limit is achieved. As shown in Fig. 2.13d, the accelerations of both parameters grow very large. This is an indication that the identified boundary corresponds to a limit that involve both parameters. Since the two parameters gradually approach each other along the geodesic path, the limit involving both parameters corresponds to the symmetry of the model, where the two parameters are equal. Evaluating this limit leads to a reduced model $x_3(t) = 2e^{-te^{log\theta_2}}$, corresponding to the bottom-right boundary (red) of the original model manifold shown in Fig. 2.12.

After presenting one geodesic path in the parameter space, natural next questions are how about its corresponding image in the data space on the model manifold, and how does the geodesic path look like given different starting points? To answer these questions, we compute initial velocities corresponding to various initial log-parameters that map to different points on the model manifold, and map the initial velocities of the log-parameters to directions on the model manifold. The mapped velocities on the manifold are visualized as a vector field on the model manifold in Fig. 2.14. From this vector field, it is easy to imagine the trajectory of the geodesic paths on this model manifold, and map out regions on the manifold that lead to each boundary.

After the first iteration of MBAM, the model reduces to $x_3(t) = 2e^{-te^{log\theta_2}}$ with only one parameter. If we perform a second iteration of MBAM with the initial log-parameter $log\theta_2 = log(11.5) = 2.4423$, the initial velocity is $+1$, and the geodesic path leads to the limit of $log\theta_2 \to \infty$. Evaluating this limit further reduces the model to a constant of $x_3 = 0$, which corresponds to the bottom-left corner of the original model manifold.

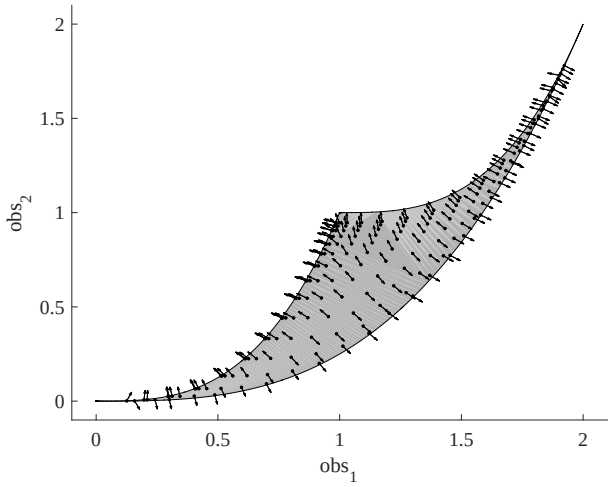


Figure 2.14: **Vector field on the model manifold** The initial velocities in the data space at the starting points of geodesic paths corresponding to initial parameters that map to various regions on the model manifold

2.3.4 Model reduction based on the profile likelihood method

The profile likelihood approach for experimental design can also be applied to perform model reduction [62]. The idea is related to the manifold boundary approximation method, but has notable differences. As mentioned in the discussions of experimental design, the profile likelihood of one parameter is a function of the parameter and is defined by solving many optimization problems with the parameter fixed at various values. The optimized values for other parameters along the profile likelihood form a trajectory in the parameter space, along a canyon defined by cross-sections of the cost surface at various values of the fixed parameter. The profile likelihood approach examines these canyons associated with the profile likelihood of all the parameters, and suggest appropriate limits to derive reduced models. Here is a simplified procedure for model reduction based on profile likelihood:

1. Compute profile likelihood for each parameter. Basically, pick one parameter and vary its value in the feasible region, and for each value perform optimization to estimate the other parameters. The optimization objective is a least squares cost function $c = 0.5 * \sum_i (obs_i - pred_i(\theta))^2$. Perform the same analysis for each parameter.

2. Define thresholds for acceptable likelihoods.

3. Examine the shapes of the profiles with respect to the threshold and decide on the limits to simplify the model. If both ends of a profile exceed the threshold, the corresponding parameter is considered to be identifiable and thus cannot be reduced. If neither ends of a profile exceed the threshold, the corresponding parameter is unidentifiable and can be fixed to an arbitrary value. If one end exceeds the threshold and the other stays below it, the corresponding parameter can be taken to the limit associated to the end that stays below the threshold.

The profile likelihood of the sum-of-two-exponential model is already presented in Fig. 2.6a and 2.6b. For both parameters, the profile likelihood exceeds the threshold on the left end, and stays below the threshold on the right end, indicating that both parameters can be taken to $+\infty$ without making the profile likelihood unacceptable. Since the example model is symmetric with respect to the two parameters, an appropriate reduction should take either one parameter to $+\infty$, or both parameters to $+\infty$. The corresponding reduced model is either $x_3(t) = e^{-\theta_2 t}$, or a constant model $x_3(t) = 0$, corresponding to the bottom-left boundary or the bottom-left corner of the original model manifold in Fig. 2.12. Both are reasonable reduced models for this example because the observed data sits close to the bottom-left corner of the original model manifold.

2.4 Conclusion and Discussion

Mathematical modeling is a crucial tool for studying complex biological processes. In systems biology, mathematical modeling often faces the situation of highly complex models and insufficient experimental data. This makes it challenging to perform parameter estimation and obtain insights into the underlying biological mechanisms that give rise to the data. In this chapter, two distinctive strategies, experimental design method and model reduction method, have been discussed. Experimental design method can help us to choose the most

informative experiment, and model reduction method can lead us to identify the key controlling part of the process. Although experimental design and model reduction have been largely considered as distinct problems in the literature, these two problems share deep connections that can unified them into a common framework. Here, we focus on the model manifold method and the profile likelihood method, which can tackle both problems by exploring their connections.

In this work, both manifold perspective and profile likelihood perspective have been discussed using the the simple example model. From the model manifold perspective, we consider a mathematical model as a manifold living in a data space, and consider the observed experimental data as a point in the data space. In addition, parameter estimation can be viewed as projecting the data point onto the manifold. By examining the singularity around the projected point on the manifold, we can perform both experimental design and model reduction. Experimental design is to identify new experiments that expand the manifold and remove the singularity to reduce parameter uncertainty. Model reduction is to identify the nearest boundary, which is the nearest singularity that suggests an appropriate form of a reduced model.

From the profile likelihood perspective, we consider a mathematical model and observed experimental data together as an optimization problem. Parameter estimation techniques and sampling techniques can be used to obtain a collection of acceptable parameters, all of which fit to the data decently well. By examining this collection of acceptable parameters, both experimental design and model reduction can be performed. Experimental design examines model predictions of new experiments based on the acceptable parameters. Also, it identifies the new experiment with largest variations in the model predictions. Model reduction examines the range of the acceptable parameters. Then, it identifies which parameters can be taken to the limits and hence removed, while still maintaining a decent fit.

One key difference between the two perspectives is the computational complexity. The

profile likelihood method is more computationally expensive. Since each iteration of the profile likelihood method involves computation of N profiles. Each profile is computed by many runs of parameter optimization, with one parameter fixed at various values. The difference in the number of parameter optimization runs suggests that the manifold boundary method is more computationally efficient compared to profile likelihood. This is because that parameter optimization is typically the most time consuming operation in these analyses. On the other hand, the parameter estimations required in profile likelihood can be achieved by relatively standard optimization procedures, whereas the numerical integration of geodesics in the model manifold approach requires more sophisticated mathematical machinery and higher numerical precision. For very complex models, it is possible that the profile likelihood approach works but the model manifold approach fails because of numerical issues with the geodesic integration.

Another key difference between the two algorithms is whether parameter symmetry is considered. The profile likelihood can identify limits that involve taking individual parameters to their limits (0 or $\pm\infty$), and can also identify combinations of parameters that go to their limits together. However, profile likelihood is not able to identify limits related to symmetry in the system, for example, the limit of $\theta_1 \to \theta_2$ in the sum-of-two-exponentials example model. On the other hand, the manifold boundary approximation method is able to identify limits associated to parameter symmetry and derive the corresponding reduced models. This kind of symmetry is actually quite common in systems biology and also in other engineering practices. For instance, nearly all machine learning models in regression analysis and neural networks have lots of internal symmetries.

CHAPTER 3
QUANTIFYING THE RELATIVE IMPORTANCE OF DATA FOR IMPROVING PARAMETER ESTIMATION

3.1 Introduction

Ordinary differential equations (ODEs) are usually used to understand and describe biological processes. ODE-based models usually contain many unknown parameters, and thus the parameter estimation is an important step toward deeper understanding of the process. Parameter estimation is often formulated as a least squares optimization problem, and this formula considers all experimental data points as equally important. However, this equal-weight formulation ignores the possibility of existence of relative importance among different data points. This may lead to misleading parameter estimation results [82, 83].

Fig. 3.1 shows one example misleading parameter estimation result because of treating all data points equally important. For the Fig.3.1 example, the equal-weight cost function is used and the measurement error is ignored. In Fig. 3.1A and 3.1B, the solid curves represent the same experimentally observed data, and the dotted curves represent model predictions based on different parameter settings, A and B. In Fig. 3.1A, the dotted curve fits the steady state accurately, whereas the dynamical region of the behavior is not well captured. Since we measure the quality of the fit by the equal-weight cost function, the shaded area in-between the two curves represent the cost value. In comparison with fig. 3.1A, the parameter setting in Fig. 3.1B produces a better fit. Although the fit is slightly off at the steady state, It captures the dynamical behavior accurately. If the steady state region lasts for a long period of time, the cost value of the second parameter (Fig. 3.1B) can be the same as the cost value in (Fig. 3.1A) or even larger. This example shows that if all data points are considered equally important, the equal-weight cost function is not able

to distinguish a poor fit from a good fit. If the weights are strategically distributed to give higher emphasis to data points in the dynamic region and lower emphasis to data points in the steady state region, the weighted cost function will be able to favor the parameter setting in fig. 3.1B over that in fig. 3.1A, regardless of the length of the steady state region. From

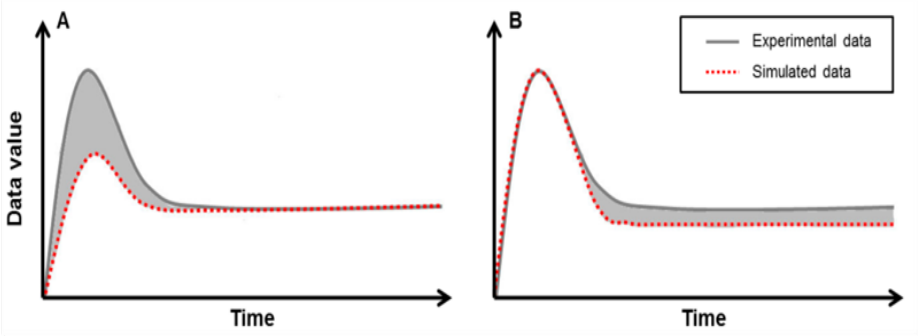


Figure 3.1: **Illustrative example showing limitations of the equal-weight cost function**

this example, we can see that weights representing the relative importance of different data points are needed (Eq. 3.1 when formulating the least squares optimization problem.

$$Cost = \frac{1}{2} \sum_{i=1}^{n} w_i (obs_i - pred_i(\theta))^2 \tag{3.1}$$

Each weight can be defined by the uncertainty of one data point given the other data points. If one data point can be accurately inferred given the other data, the uncertainty of this data point is low and the importance of this data point is low. Whereas, if inferring one data point from the other data is almost impossible, it contains a huge uncertainty and carries more information for estimating parameters. In order to consider the relative importance of data points when formulating the parameter estimation cost function (Eq.3.1), we need to define a weight for each data point by the amount of unique information it provides [82, 83]. In other words, we define each weight of a data point by the uncertainties of the data point given thel other data points. This uncertainty quantifies how well we can infer one data point using the other data points. For example, if one data point can be accurately predicted given the other data points, its uncertainty is low, and it carries a very small

amount of new information beyond what other data points provide. On the other hand, if one data point cannot be accurately inferred given the other data points, this data point has high uncertainty and carries unique information beyond other data points.

3.2 Methods

3.2.1 Parameter uncertainty given experimental data

Before explaining how we define the weights given the other data, we first discuss the uncertainty of parameters given data points. Assume we have an optimal parameter setting ($\theta^\star$) which minimizes the weighted cost function (Eq. 3.1). A small region of parameter settings near the optimal parameter set exists. These near optimal parameter settings form the confidence interval for the estimated optimal parameter, and the corresponding variation in these near optimal parameters is the uncertainty of parameters, which can be estimated by a second-order Taylor expansion of the cost function (Eq.3.1) at the optimal parameter set.

$$\begin{aligned} C(\theta) &\approx C(\theta^\star) + \frac{1}{2}\sum_{a=1}^{m}\sum_{b=1}^{m}\frac{\partial^2 C}{\partial\theta_a\partial\theta_b}(\theta_a-\theta_a^\star)(\theta_b-\theta_b^\star) \\ &= C(\theta^\star) + \frac{1}{2}(\theta-\theta^\star)^T H(\theta-\theta^\star) \end{aligned} \tag{3.2}$$

In Eq. 3.2, m represents the number of parameters. The first-order term does not appear in the Taylor expansion because the gradient of the cost function at the optimal parameter is 0, and the second-order derivatives evaluated at the optimal parameter is the Hessian (H) at the optimal parameter. The eigenvalues and eigenvectors of the Hessian matrix reflect the confidence interval of the near optimal parameters. For instance, when the Hessian has a small eigenvalue, moving the optimal parameter along the corresponding eigenvector direction does not significantly increase the cost value, leading to a huge confidence interval and a large uncertainty of the optimal parameter. On the other hand, if the eigenvalues of

the Hessian are all large, moving the optimal parameter in any eigendirection will lead to large increase in the cost value which means a very small confidence interval and small uncertainty of optimal parameter. The Hessian matrix can be approximated by the Fisher Information Matrix (FIM) as follows (Eq.3.3),

$$\begin{aligned} H_{a,b} &= \frac{\partial^2 C}{\partial\theta_a \partial\theta_b}\Big|_{\theta^\star} \\ &= \sum_{i=1}^{n} \frac{\partial}{\partial\theta_b}\left((obs_i - pred_i(\theta))(-w_i)\frac{\partial pred_i(\theta)}{\partial\theta_a}\right)\Big|_{\theta^\star} \\ &= \sum_{i=1}^{n} \left(w_i \frac{\partial pred_i}{\partial\theta_a}\frac{\partial pred_i}{\partial\theta_b} - w_i(obs_i - pred_i(\theta))\frac{\partial^2 pred_i}{\partial\theta_a\partial\theta_b}\right)\Big|_{\theta^\star} \\ &\approx \sum_{i=1}^{n} w_i \frac{\partial pred_i}{\partial\theta_a}\frac{\partial pred_i}{\partial\theta_b}\Big|_{\theta^\star} \end{aligned} \tag{3.3}$$

In Eq. 3.3, n represents the total number of data points, and in the final line, the approximation is based on the assumption that the fit error is very small at the optimal parameter. This approximation of Hessian is the Fisher Information matrix, and its inverse is the covariance matrix that approximates the uncertainty of the optimal parameter given the data. The parameter uncertainty can be quantified using Eq. 3.4,where m is the number of parameters and I is the Fisher Information matrix.

$$Uncertainty(\theta|data) = \frac{1}{m} trace(I^{-1}) \tag{3.4}$$

Since model parameters in biology models are often constrained to be non-negative, it is often advantageous to compute the Fisher Information matrix in the log-parameter space: $I_{a,b} = \sum_{i=1}^{n} w_i \frac{\partial pred_i}{\partial \log(\theta_a)} \frac{\partial pred_i}{\partial \log(\theta_b)}|_{\theta^\star}$. In addition, the Fisher Information matrix can be represented by $J^T J$, where the J represents the Jacobian matrix. Therefore, the eigenvalues of Fisher Information matrix are equal to the squares of the singular values of the Jacobian, and the Eq.3.4 can be calculated by $\sum_{a=1}^{m} \frac{1}{{s_a}^2}$, where s indicates the singular values of the Jacobian.

3.2.2 Data uncertainty given other data

Similar to the formulation of parameter uncertainty given data, the uncertainty of estimating one set of data points (S_1) given another set of data points (S_2) using the Fisher Information can be defined as follow:

$$Uncertainty(dataS_1|dataS_2) = \frac{1}{m} trace(I_{S1} {I_{S2}}^{-1}) \tag{3.5}$$

where $[I_{S1}]_{a,b} = \sum_{i \in S1} w_i \frac{\partial pred_i}{\partial \log(\theta_a)} \frac{\partial pred_i}{\partial \log(\theta_b)}|_{\theta^\star}$, and $[I_{S2}]_{a,b} = \sum_{i \in S2} w_i \frac{\partial pred_i}{\partial \log(\theta_a)} \frac{\partial pred_i}{\partial \log(\theta_b)}|_{\theta^\star}$. $\theta^\star$ here is the best fit parameter defined by data points in S_2. The matrix inverse and multiplication inside the $trace$ operation in Eqn. 3.5 approximate the derivatives of data points in S_2 with respect to data points in S_1. To quantify the importance of a data point, we calculate the uncertainty of one data point i given all the other data points. We define the two subsets as follows: S_1=$\{i\}$ and S_2=$\{1, 2, ..., n\} \backslash S_1$. The Fisher Information matrices I_{s1} and I_{s2} are computed using the i^{th} row of the Jacobian and all the other $(n-1)$ rows of Jacobian, respectively. The data uncertainty reflects whether one data point can be accurately predicted based on all other data points. As shown in Eq.3.5, calculating the weight of a data point requires the Fisher Information matrix evaluated at the optimal parameter, which is in turn defined by optimizing the weighted least squares cost function (Eq.3.1) that requires the weights.

3.2.3 Iterative algorithm for quantifying the weight of each data point

Fig.3.2 shows a flowchart of the iterative algorithm for quantifying the different weight of each data point. In the first iteration, the algorithm is initialized by assigning equal weights, 1, to all data points. To optimize the parameters with respect to the initial equal-weight cost function, the interior-point algorithm [84] is used with randomly generated initial parameters. We evaluate the Jacobian at the estimated parameter, which is used for calculating the uncertainty associated to each data point (Eq.3.5). The uncertainty of each

data point serves as its updated weight. We normalize the weights, so that the sum of the weights equals the total number of data points. Such a normalization makes the weighted cost function and the equal-weight cost function comparable. In the subsequent iterations,

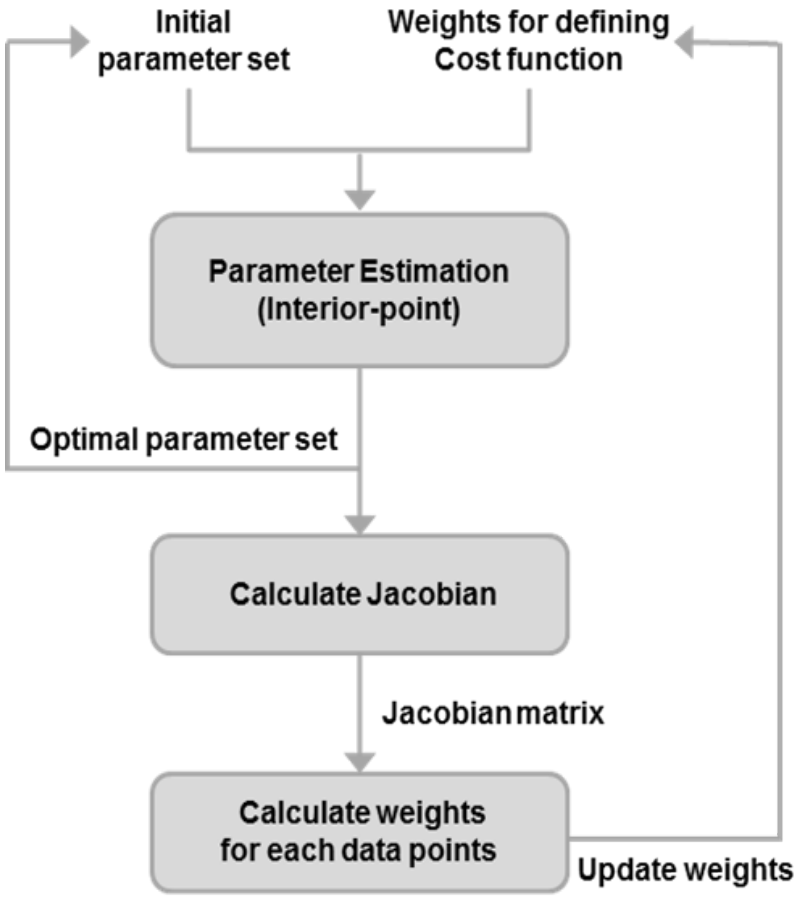


Figure 3.2: **Flowchart of the iterative algorithm for quantifying the weight of each data point**

the estimated parameter and the updated weights from the previous iteration serve as the initial parameter and weights for the parameter estimation step. Therefore, the parameter estimation is performed with respect to the updated weights. After that, the weights are re-computed based on the newly estimated parameter. This process is repeated until the weights converge.

The intuition of this algorithm is that even if the optimal parameter setting derived from the equal-weight cost function in the first iteration is incorrect, as long as it represents a

decent fit to the data, the updated weights at the end of the first iteration will roughly capture the curvature and relative importance of the data points. The subsequent iterations start with the updated weights, and will gradually adjust and fine-tune the optimal parameter, as well as the weights until convergence. In practice, to ensure the first iteration obtains a decent fit, we typically perform 100 runs of parameter estimation using random initial parameters generated by the Latin hypercube sampling method. We pick the best estimated parameter among the 100 to compute the Jacobian and updated weights. The second and subsequent iterations pursue the best fit in the first iteration and fine-tune it until weights converge, which typically takes only a few iterations.

3.2.4 Sampling algorithm

To evaluate the performance of the re-formulated weighted cost function and compare with the equal-weight cost function, we developed a parameter sampling algorithm, similar to the Markov Chain Monte Carlo algorithm [85]. The sampling algorithm generates a collection of near optimal parameter settings. Given an acceptance threshold for near optimal, the sampling algorithm identifies the acceptable parameter region which is defined as the union of all parameter settings whose cost value is smaller than the given acceptance threshold. Although most of the acceptable parameter settings are not optimal, they generate decent model predictions which fit well to the data. Thus, this acceptable parameter region can be used to visualize the confidence interval of the estimated parameter and the model predictions. Fig. 3.3 shows a flowchart for the sampling algorithm. It starts with the estimated parameter setting obtained from optimizing the weighted cost function, which serves as a seed (θ^{curr}) inside the acceptable parameter region. To explore the parameter space, we perturb the current parameter, and we evaluate the cost function at the perturbed parameter. If the cost value is smaller than the acceptance threshold, the perturbed parameter is accepted and becomes the current parameter. On the other hand, if the cost value is larger than the threshold, the perturbed parameter is rejected and the current parameter is

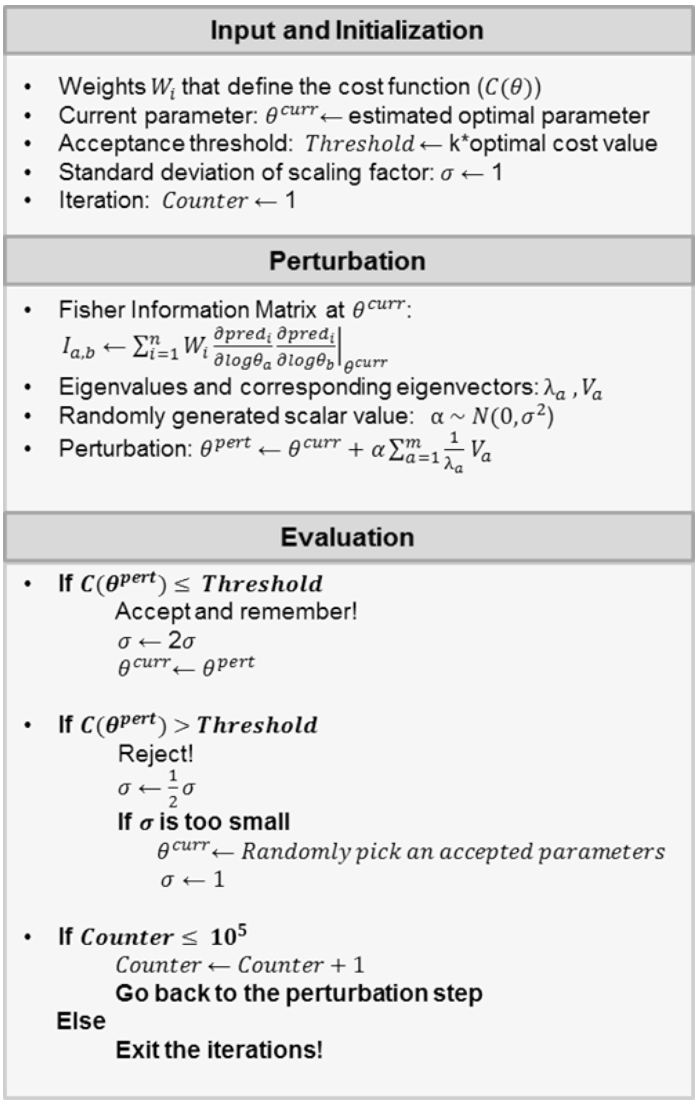


Figure 3.3: **Flowchart of the parameter sampling algorithm**

not changed. This sampling algorithm is performed iteratively, resulting in a collection of acceptable parameters.

In order to achieve efficient sampling and low rejection rate, we designed the direction and amplitude of the perturbation using the Fisher Information Matrix and parameter uncertainty. At each iteration of the sampling algorithm, we compute the inverse of the Fisher Information matrix at the current parameter. We apply larger perturbation along the eigendirection associated to large eigenvalues, and smaller perturbation along the eigendirection associated to small eigenvalues. Since the inverse of the Fisher Information Matrix

approximates the covariance of the estimated parameter, such a choice of perturbation leads to larger perturbations along the insensitive directions that have little influence on the cost function, and smaller perturbations along the sensitive direction, enabling efficient exploration even when the covariance is highly anisotropic.

To determine the amplitude of the perturbation, a normal distribution, $N(0, \sigma^2)$ is used. A large σ value makes the sampling algorithm explore the parameter space quickly, but is at the risk of low acceptance rate and low sampling efficiency. On the other hand, a small σ value has the opposite effect. In the sampling algorithm, the value of σ is adjusted during the process, doubled when the perturbed parameter is accepted and halved when the perturbed parameter is rejected. The purpose of adjusting σ is to balance between the exploration and the acceptance rate. Furthermore, if the σ value is too close to zero, meaning that the sampling process is stuck at a narrow corner of the acceptable parameter region, we randomly pick a previously found acceptable parameter and set is as the current parameter. This heuristic effectively resets the sampling process when it is stuck.

3.3 Results

3.3.1 G1/S transition module

To test the iterative algorithm for computing uncertainty-based weights, the G1/S transition model was used [86, 87]. This model consists of 2 variables, pRB (Retinoblastoma protein) and E2F1 (Activator), and 10 model parameters. We also consider the initial concentrations of both variables as parameters, and therefore, the total number of model parameters is 12. The ordinary differential equations of the model are described in Eq. 3.6.

$$\begin{aligned} \tfrac{d}{dt}[pRB] &= K_1 \tfrac{[E2F1]}{K_{n1}+[E2F1]} \tfrac{J_{11}}{J_{11}+[pRB]} - \varphi_{pRB}[pRB] \\ \tfrac{d}{dt}[E2F1] &= K_p + K_2 \tfrac{a^2+[E2F1]^2}{K_{n2}^2+[E2F1]^2} \tfrac{J_{12}}{J_{12}+[pRB]} - \varphi_{E2F1}[E2F1] \end{aligned} \tag{3.6}$$

To generate the ”experimental data” shown in Fig. 3.4, we simulated Eq. 3.6 using the parameter setting in [86] as the true parameter, with observed time points evenly spaced every 25 minutes from 0 to 800 minutes, and therefore, the observed data points are evenly spaced along the time axis.

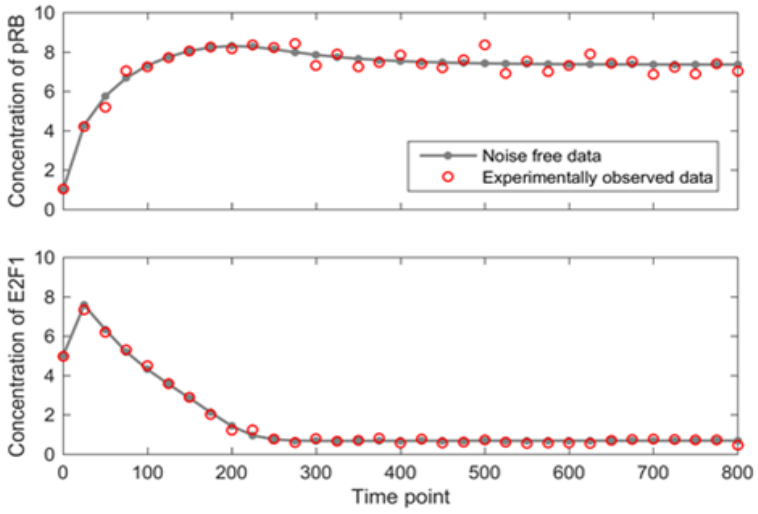


Figure 3.4: **Experimental data of the G1/S transition model** The gray curve: simulated noise free data obtained from true parameters, The red circles: noisy data (adding and multiplying a small amount of Gaussian noise)

G1/S transition module with 6 parameters

For the first model, we examined a simple setting where only 6 parameters are unknown and need to be estimated. The 6 unknown parameters are the initial condition of two variables $(pRB_(t = 0), E2F1_(t = 0))$ and 4 model parameters K_1,J_{11}, K_{n2}, and J_{12}.

The weights are shown in Fig. 3.5. The two data points at t=0 received the largest weights because they are directly related to the unknown parameters. In general, data points corresponding to the dynamic time regions received higher weights compared to data points corresponding to the flat regions. After obtaining the weights, we compared the equal-weight cost function and the weighted cost function, using the sampling algorithm. For each cost function, we used the interior point algorithm to obtain a best fit. We defined

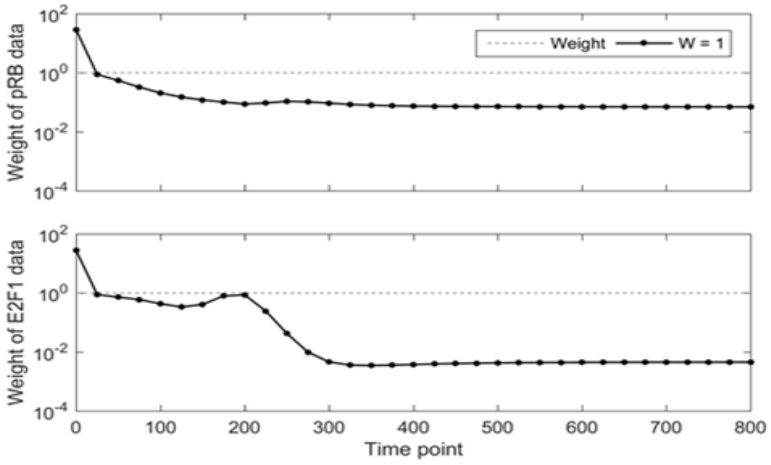


Figure 3.5: **Weights of the G1/S transition with 6-parameters** The black circle: weight of each data point on the log scale, the dashed line: weight of the equal-weight cost function

the acceptable parameter region as the collection of parameters whose cost function value is smaller than 3 times the cost of the best fit. The sampling algorithm was used to obtain a collection of acceptable parameters. Finally, we simulated the model using the acceptable parameters, and overlaid the model simulations with the experimental data. The resulting visualization is a simulated belt centered around the experimental data, showing the range of model predictions based on the acceptable parameters.

The first column of Fig. 3.6 visualizes the acceptable parameters derived from the equal-weight cost function. In the bottom panel for E2F1, we can see that the belt is quite thick in the dynamic region of the data and quite thin in the flat region. This is because the large number of data points in the flat region all reflects the steady state, forcing the parameter optimization algorithm to focus on finding the steady state accurately, even at the expense of errors in the dynamic region. The unbalanced belt width is a manifestation of the limitations of the equal-weight formulation as discussed in Fig. 3.1 in the Introduction section. The unbalanced belt width is less pronounced in the upper panel for pRB, because the flat region is shorter, and the data is noisier when pRB reaches its steady state.

The second column of Fig. 3.6 visualizes the acceptable parameters derived from the

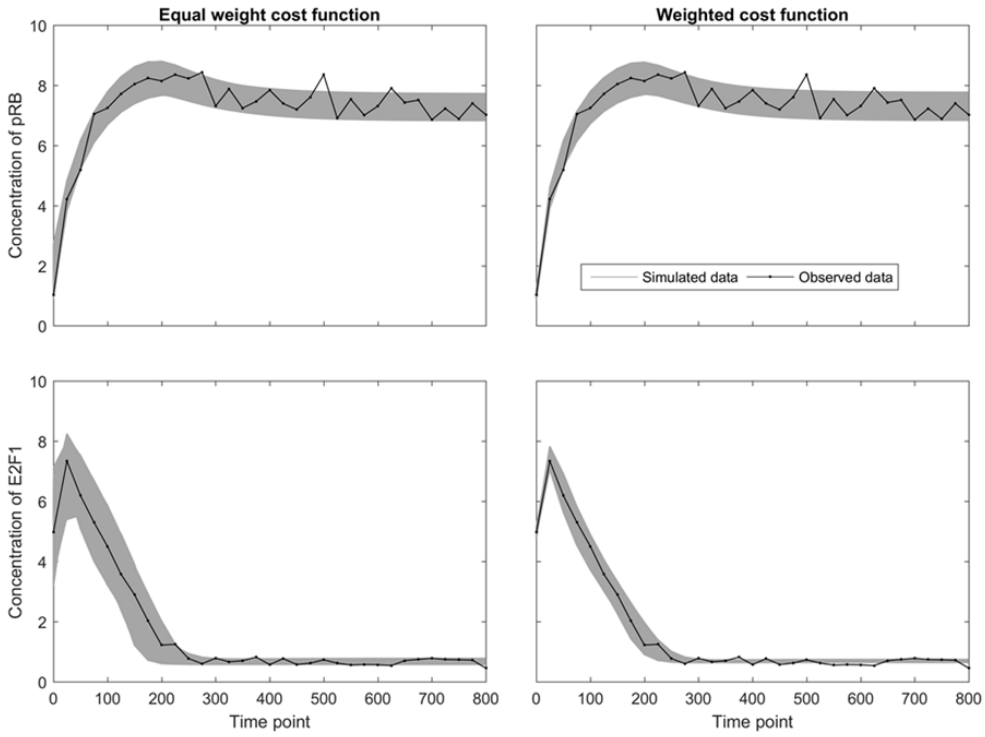


Figure 3.6: **Numerical result for evaluating G1/S transition with 6-parameters** The black curve: experimental data, The gray curve: simulated data sets obtained from the sampling algorithm.

weighted cost function. In the bottom plot, we can see that belt is much thinner in the dynamic region, compared to the first column of Fig. 3.6. This is because data points corresponding to the dynamic region received large weights while data points in the flat region received small weights. Since the data points at t=0 received very large weights, the beginning of belts are extremely thin, indicating that the two initial condition parameters are estimated with high accuracy. Overall, the comparison in Fig. 3.6 shows that weights computed by the iterative algorithm is able to discount redundant information present in the data, and enable the parameter optimization step to better capture the dynamic behavior in the data.

To test the sensitivity of the algorithm for computing the weights, we generated 100

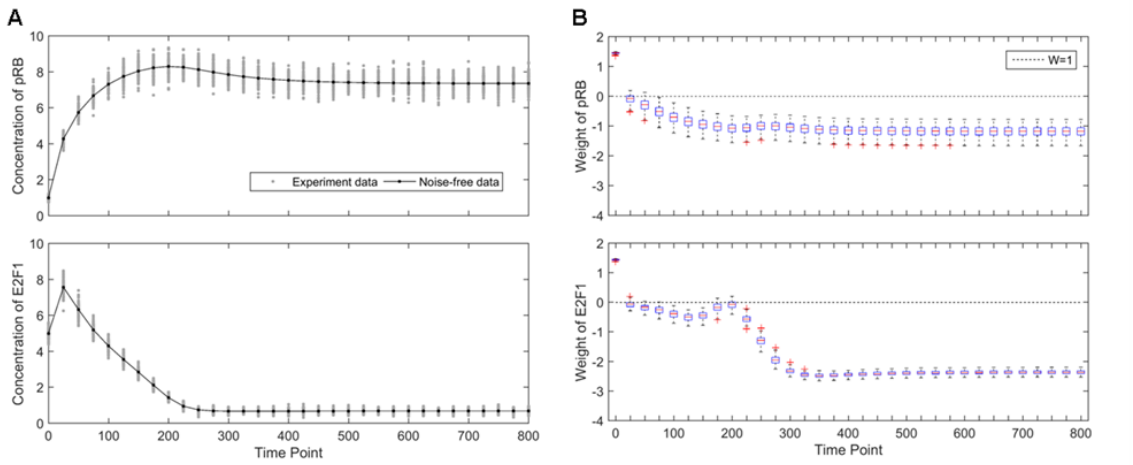


Figure 3.7: **G1/S transition with 6-parameters: robustness of the uncertainty-based weights** A) The black curve represents the noise-free data and the gray dots represent 100 simulated noisy data sets for sensitivity analysis. B) The dotted line represent the weight of equal weight cost function ("0" on log scale), and each box represents the weights for one data point, computed from the 100 noisy experimental datasets. The small range of each box indicates the robustness of the uncertainty based weights.

experimental datasets by randomly perturbing the noise-free simulation in Fig. 3.4. The variation among the 100 datasets is shown in Fig. 3.7A. Using the iterative algorithm, weights are computed based on each experimental dataset. The variation among the 100 sets of weights is shown in Fig. 3.7B. The first measurement time point for both variables consistently receive large weights across the 100 datasets, very robust to the noise. For other measurement time points, we can observe the same pattern as in Fig. 3.5, where the dynamically change regions consistently receive higher weights than flat regions.

G1/S transition module with 12 parameters: evenly spaced along the time axis

The experimental data is shown in Fig. 3.4, the gray curve represents a noise free data obtained from the true parameters and the red circles represent the noisy data which will be used as experimental data. The simulated data was perturbed with a small amount of multiplicative and additive Gaussian noise [24]. Since pRB inhibits E2F1 activation, the concentration of E2F1 decreases as the concentration of pRB increases as shown in Fig. 3.4. Afterwards, the concentrations of both variables approach steady state gradually.

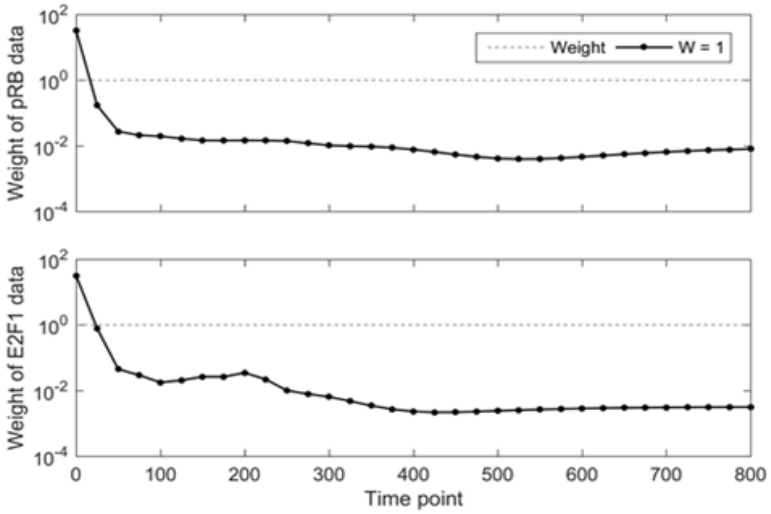


Figure 3.8: **Weights of the G1/S transition with 12-parameters** The black circle: weight of each data point on the log scale, the dashed line: weight of the equal-weight cost function

The iterative algorithm is carried out with this data to calculate corresponding weights. The resulting weights are shown in Fig. 3.8 : larger weights in dynamic regions and lower weights in flat regions. As shown in Fig. 3.8, the resulting weights are qualitatively similar to Fig. 3.5, but not exactly the same. This analysis shows that the weights do not just simply encode the curvature of the data. They are also dependent on the mathematical structure of the model and which parameters need to be estimated. Another example not shown here is that: if we assume the initial conditions of the two variables are known and only aim to estimate the remaining 10 parameters, the data points at t=0 will receive weight 0. This is because the data points at t=0 represent noisy measurements of the initial condition. When the initial conditions are known, the data points at t=0 do not provide any new information for estimating the other parameters. This example again shows that the weights are also influenced by the mathematical structure of the model.

Fig. 3.9 illustrates the model predictions of acceptable parameters for both cost functions, using the sampling algorithm. The equal-weight cost function led to highly unbalanced belt width because of the large number of data points in the steady state region, whereas the weighted cost function led to relatively balanced belt width by assigning larger

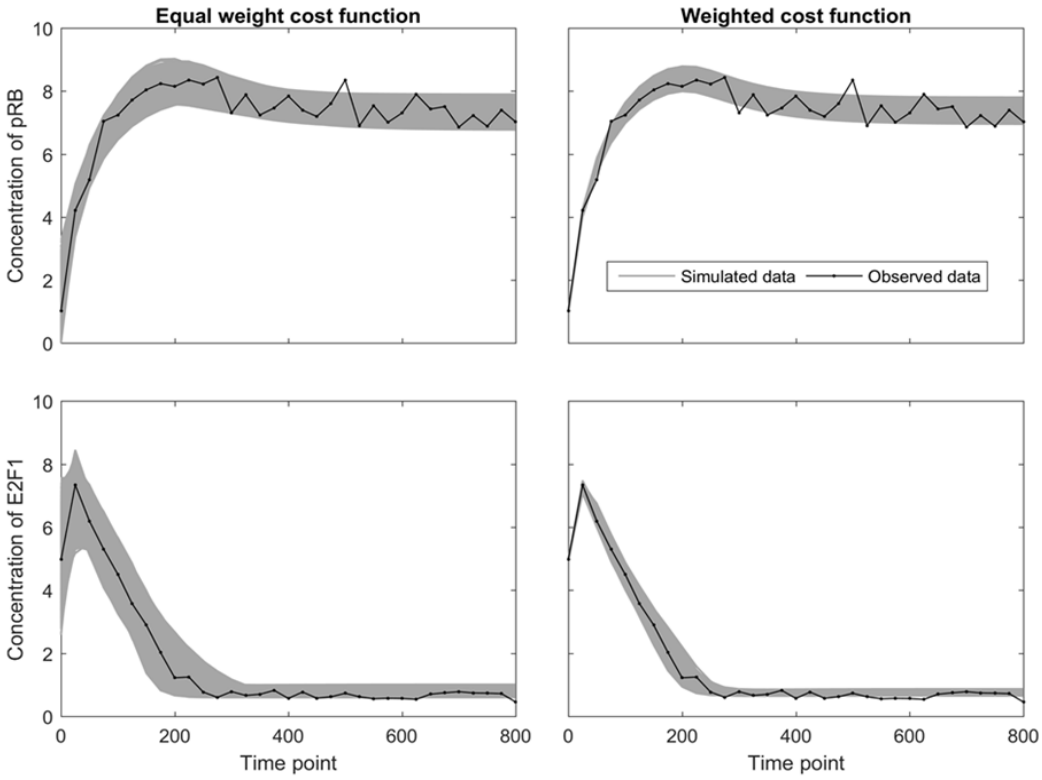


Figure 3.9: **Numerical result for evaluating G1/S transition with 12-parameters** The black curve: experimental data, The gray curve: simulated data sets obtained from the sampling algorithm.

weights to data points in dynamic regions and lower weights to the data points in flat regions.

Similar as before, we compared the equal-weight cost function and the weighted cost function for this 12-parameter model. As shown in Fig. 3.9, the comparison is qualitatively the same as the previous 6-parameter model. The equal-weight cost function led to belt with highly unbalanced width, favoring the steady state. The weighted cost function focused more on the dynamic region of the data, generating well constrained model predictions (thin belt width) for the early time points.

To examine the sensitivity of the weights in the 12-parameter model, we used the same

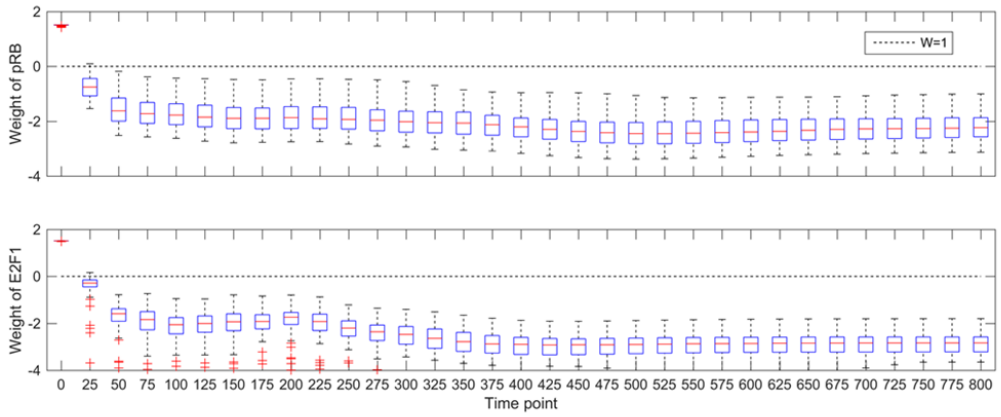


Figure 3.10: **G1/S transition with 12-parameters: robustness of the uncertainty-based weights** The dotted line represent the weight of equal weight cost function ("0" on log scale), and each box represents the weights for one data point, computed from the 100 noisy experimental datasets.

100 experimental datasets introduced in the previous example (Fig. 3.7A), and calculated weights based on the 100 datasets. The variation of the weights is shown in Fig. 3.10. Comparing Fig. 3.7B and Fig. 3.10, we can see that weights of the data points in this 12-parameter model are not as robust as the weights in the previous 6-parameter model. This is caused by the higher complexity of the 12-parameter model, making the parameter estimation component of the iterative algorithm to overfit to the noise and subsequently lead to different weights.

G1/S transition module with 12 parameters: unevenly spaced along the time axis

In the 12-parameter models above, the experimental data points are evenly spaced along the time axis. As shown in Fig.3.8, data point in different time periods receive different weights. The magnitudes of weights decrease as time increases, indicating that the data points located at later time points may be redundant. In order to reduce the effect of these redundant data points, we manually selected unevenly spaced time points as the experimental observations: data points in dynamic regions are densely sampled, while data points in steady state region are sparsely sampled. The selected time points are

$0, 5, 10, 15, 20, 30, 40, 50, 75, 100, 125, 150, 175, 200, 300, 400, 600,$ and 800. The experimental data with these time points is shown in Fig. 3.11.

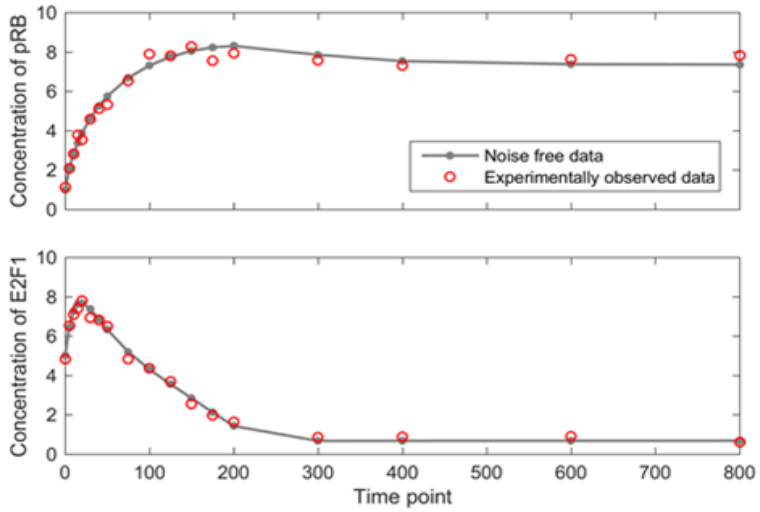


Figure 3.11: **Experimental data of the G1/S transition model with unevenly spaced time points** The gray curve: simulated noise free data obtained from true parameters, The red circles: noisy data (adding and multiplying a small amount of Gaussian noise)

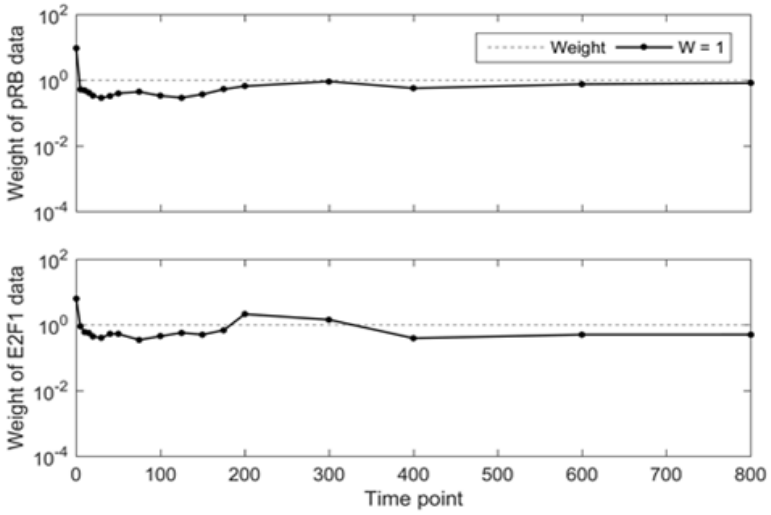


Figure 3.12: **Weights of the G1/S transition with unevenly spaced time points** The black circle: weight of each data point on the log scale, the dashed line: weight of the equal-weight cost function

The circles represent the noisy data at the unevenly measurement time points and the

gray curve represents simulated noise free data obtained from true parameters. Using this data, the iterative algorithm was performed to obtain the weights shown in Fig. 3.12. In contrast to the weights in the previous examples, all data points except the first data point (t = 0) receive very similar weights regardless of the region (dynamic or steady state). This is because the measurement time points are selected strategically to make the data points roughly equally important, so that redundancy among the data points is reduced.

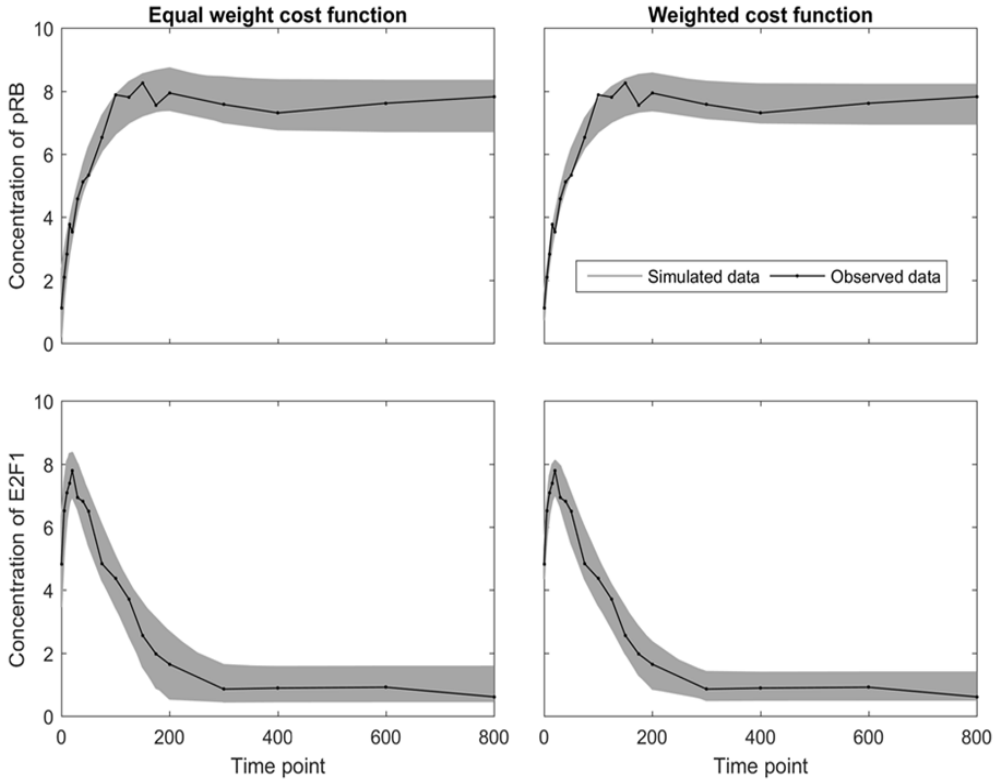


Figure 3.13: **G1/S transition with unevenly spaced data: Results of the sampling algorithm** The black curve: experimental data, The gray curve: simulated data sets obtained from the sampling algorithm.

Using the sampling algorithm, the model predictions of acceptable parameters for these two cost functions are visualized in Fig.3.13. The belts associated to the two cost functions are quite similar to each other. This is due to the low variation among the weights shown

in Fig. 3.12, which makes the weighted cost function and the equal-weight cost function almost equivalent to each other. This example shows that the weighted cost function can also be achieved by strategically selecting measurement points to avoid redundancy in the resulting experimental data.

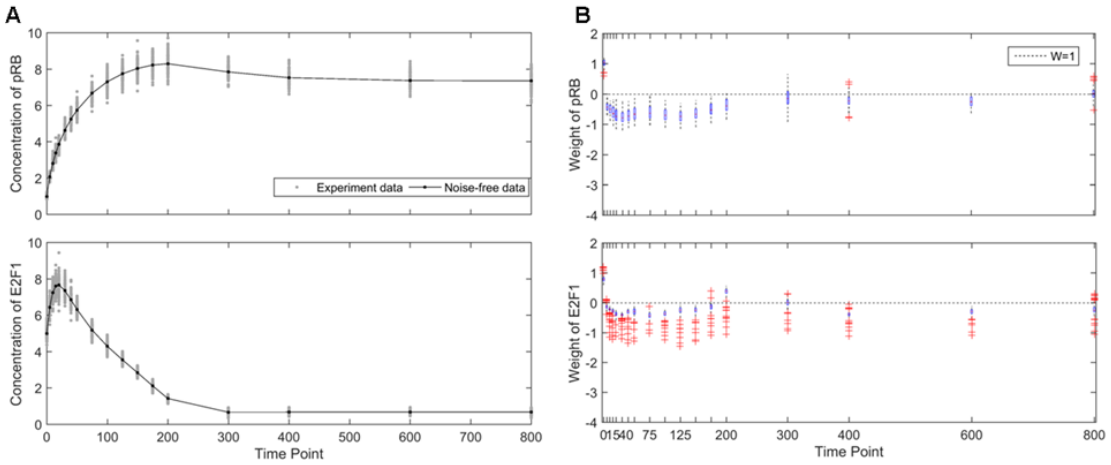


Figure 3.14: **G1/S transition with unevenly spaced time points: robustness of the uncertainty-based weights** A) The black curve represents the noise-free data and the gray dots represent 100 simulated noisy data sets for sensitivity analysis. B) The dotted line represent the weight of equal weight cost function ("0" on log scale), and each box represents the weights for one data point, computed from the 100 noisy experimental datasets. Although some outliers exist, majority of the weights are very close to the dotted line, meaning that the weights are robust to the simulation noise.

To test the sensitivity of the iterative algorithm, we randomly simulated 100 experimental datasets, shown in Fig. 3.14A. We applied the iterative algorithm to compute weights based on each experimental dataset. Fig. 3.14B describes the variations among 100 sets of weight, where weights for the majority of the simulated datasets are close to "1" (0 in log scale), corresponding to the equal-weight cost function.

3.3.2 MAPK module

To test the weighted cost function in a more complex model, the MAPK module was considered [88]. This module consists of 5 variables and 9 model parameters. The ordinary differential equations of this module are depicted in Eq. 3.7, where X represents MAPK,

XE represents the complex of X with the enzyme E, XP is singly phosphorylated form of MAPK, XPE is complex of XP with the enzyme, and XPP is doubly phosphorylated form [88]. In this example, we assume that the concentration of the enzyme E is initially 0.01, changes to 10 at t=1, and changes back to 0.01 at t=5. When the enzyme concentration is increased at t=1, the dynamic variables either increase or decrease, in respond to the enzyme change. To generate the experimental data, we used the model parameters in [88] as the true underlying parameters. All 9 parameters and 5 initial conditions were considered as unknown model parameters, thus the total number of parameters is 14.

$$\begin{aligned}
\tfrac{d}{dt}[X] &= -K_1[X]E + K_2[XE] + K_7[XP] \\
\tfrac{d}{dt}[XE] &= K_1[X]E - (K_2 + k_3)[XE] \\
\tfrac{d}{dt}[XP] &= K_3[XE] - K_7[XP] - K_4[XP]E + K_5[XPE] + K_8[XPP] \\
\tfrac{d}{dt}[XPE] &= K_4[XP]E - (K_5 + K_6)[XPE] \\
\tfrac{d}{dt}[XPP] &= K_6[XPE] - K_8[XPP]
\end{aligned} \tag{3.7}$$

Fig. 3.15 shows the simulated noise-free data generated from the true parameter and noisy experimental data obtained by randomly perturbing the noise-free data. The measurement time points are evenly spaced, every 0.5 hours from 0 to 10 hours. Therefore, the total number of experimental data is 105 (21 data points for each variable). When the catalyzing enzyme E concentration was changed at t=1 and t=5, the dynamic variables responded to the change. For example, the concentration of X decreased rapidly after t=1, while the concentrations of remaining four variables increased. As the catalyzing enzyme E decreased back at t=5, all variables returned to the initial condition gradually.

Using the iterative algorithm, we calculated weights for the data points, shown in Fig. 3.16. Similar to the previous models, the initial data point at t=0 receives the largest weight because it is directly related to some of the unknown model parameters. The data points

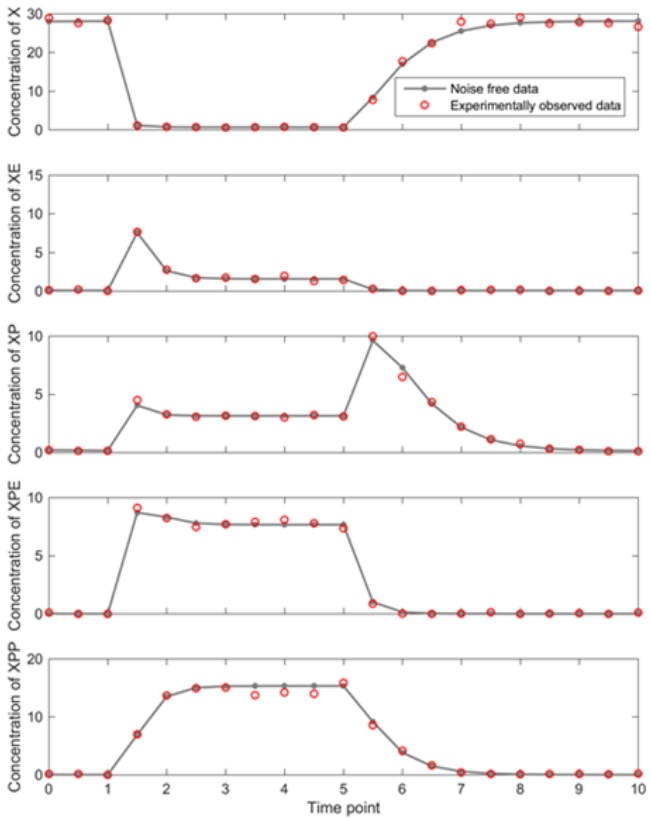


Figure 3.15: **Experimental data of the MAPK module** The solid curve represents noise-free data obtained from the true parameter. The circles represent the noisy experimental data, generated by adding and multiplying a small amount of Gaussian noise.

in the dynamic region (from t=1 to t=2) of all five variables receive the large weights. Since all five variables exhibit little dynamics from t=2 to t=5, weights of data points in these flat regions are relatively small. After t=5, weights of X, XP, XPE, and XPP slightly increased because their model predictions are changed dynamically, whereas the concentration of the XE barely changed and hence its data points after t=5 received small weights.

Fig. 3.17 shows the results of the sampling algorithm for both cost functions. In this example, the acceptance threshold was defined as five times the cost value of the optimal parameter setting. In the first column, equal-weight cost function, the belt of variable XE

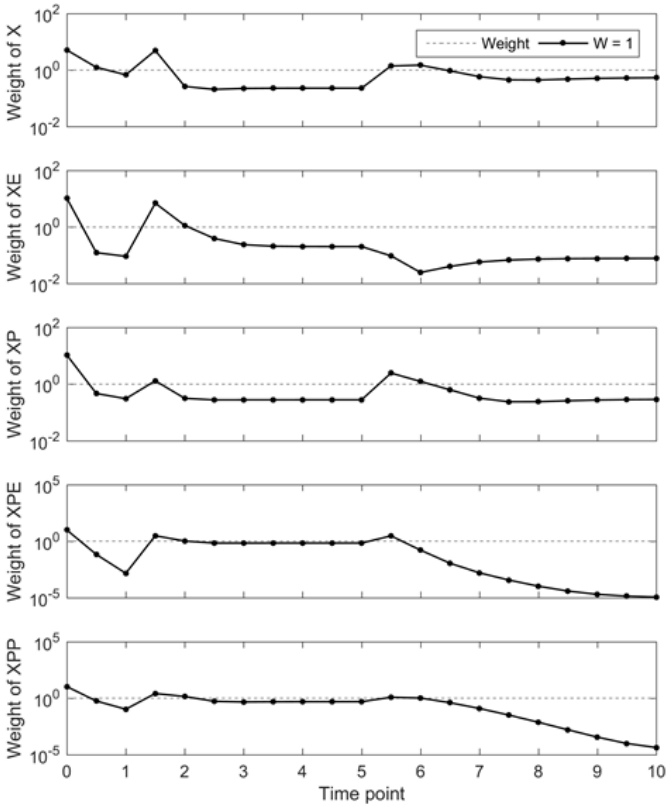


Figure 3.16: **Weights of the MAPK module** Each dot represents the weight of a data point, and the dashed line corresponds to the equal-weight cost function. Data points in dynamically changing regions receive larger weights and the data points in flat regions receive relatively smaller weights.

(the second row) is quite thick in the dynamic region and thin in the flat region. This imbalanceness of acceptable model predictions between dynamic and flat regions is consistent with results in the previous examples. In the second column, the corresponding belt of variable XE generated with the weighted cost function is much thinner in the dynamic region, compared to the equal-weight cost function. This is because the dynamic regions receive larger weights than the flat regions. For all five dynamic variables, the belt width (variation in acceptable model predictions) from the weighted cost function is smaller than or equal to that from the equal-weight cost function.

To test the noise sensitivity of the weights in the MAPK module, we randomly gener-

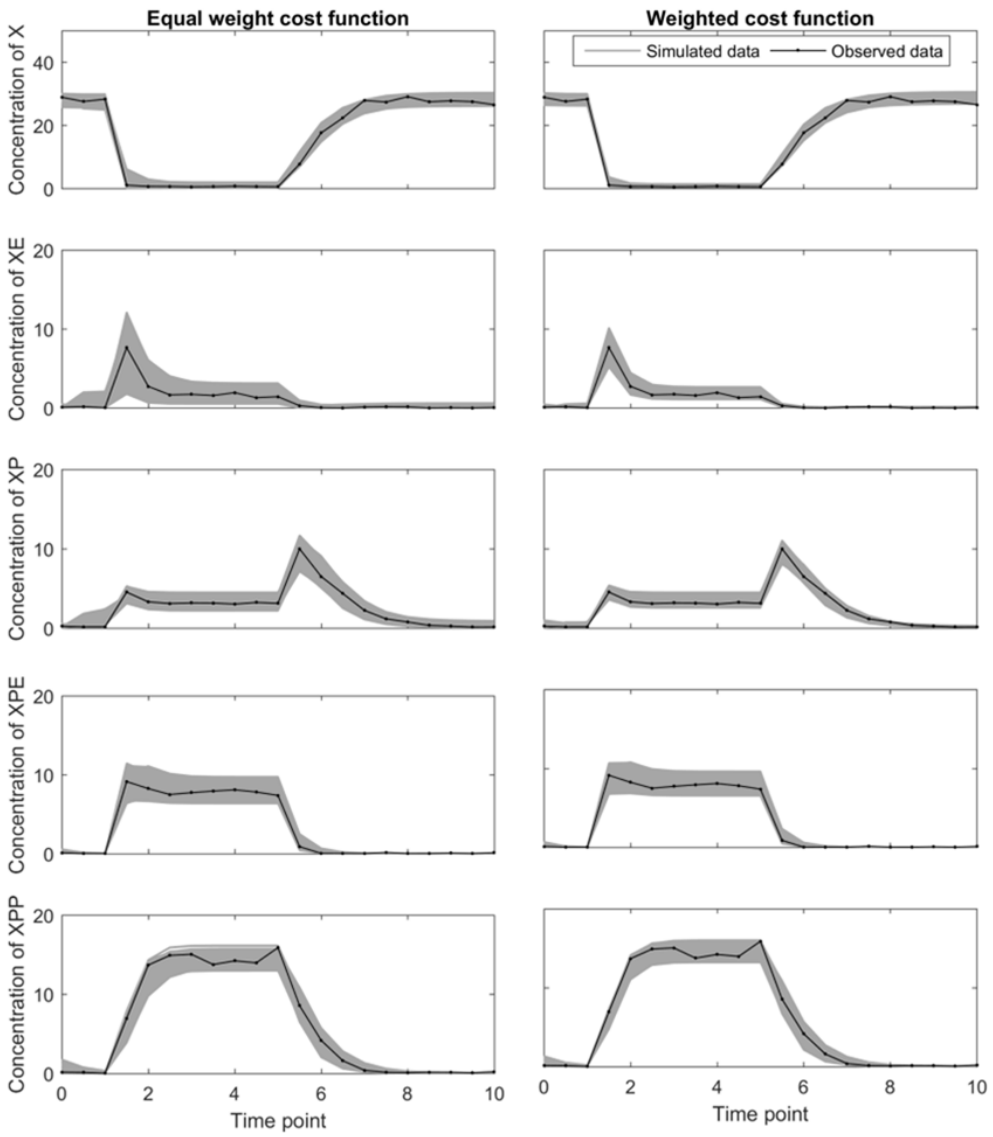


Figure 3.17: **Numerical result for evaluating the MAPK module** The black curves show the noisy experimental data. The gray belts show the model predictions based on the acceptable parameters obtained by the sampling algorithm. By comparing the belt width of the second variable XE between the two cost functions, we can see the benefit of the weight cost function. The equal-weight cost function generates imbalanced belt width between dynamic regions and flat regions. The weighted cost function produces a thin belt, meaning that it is able to better constrain the model parameters to reproduce the experimental data.

ated 100 noisy time series data, as shown in Fig.3.18. We applied the iterative algorithm to compute weights starting from each of the 100 times series data. The resulting weights are

shown in Fig.3.19. The first measurement time points of all variables consistently receive high weight, because they directly reveal the initial condition parameters. The variation of each weight across the 100 noisy datasets is small compared to the variation of weights across different data points, indicating robustness of the algorithm with respect to noise.

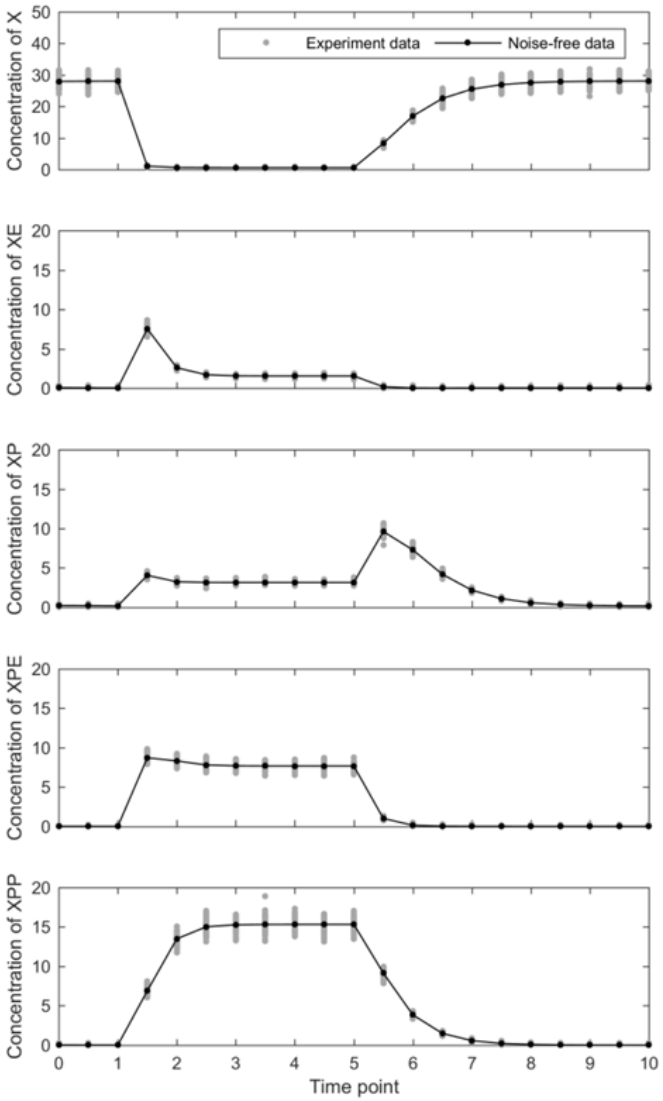


Figure 3.18: **Simulated datasets for MAPK module** The black curve represents the noise-free data, and the gray dots represent 100 simulated noisy datasets for sensitivity analysis.

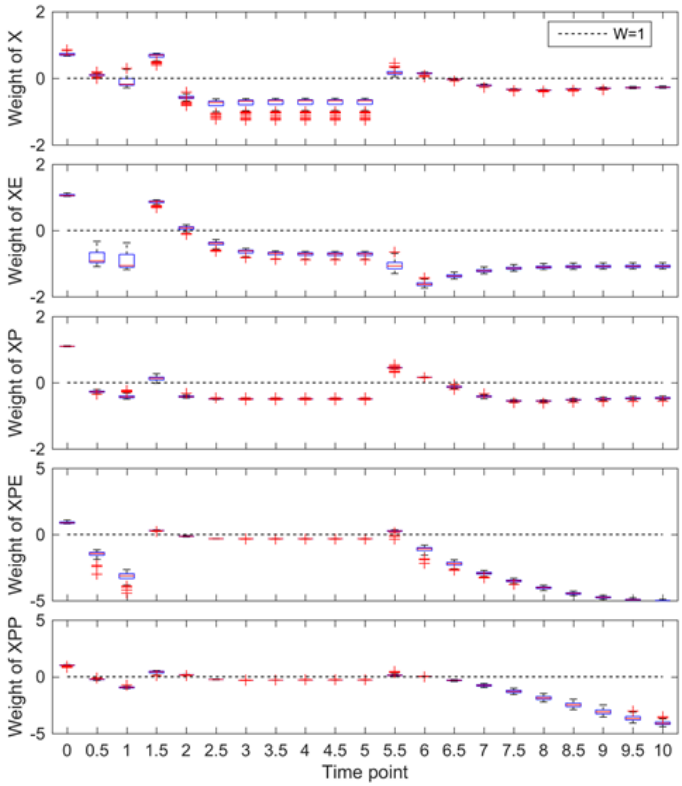


Figure 3.19: **MAPK module: robustness of the uncertainty-based weights** The dotted line indicates the equal-weight cost function. Each box shows the variation in of one weight caused by variations among the 100 noisy experimental datasets. Although some outliers exist, most weights exhibit small variations in this sensitivity analysis, showing the robustness of the uncertainty-based weights.

3.4 Conclusion and Discussion

In this chapter, we have demonstrated the new novel approach, weighted cost function which reflects the relative importance of each data point when estimating the model parameters by assigning a different weight. The weight of each data point is defined by the uncertainty of each data point given the other data points which quantifies the amount of unique information it carries. More specifically, high weights are assigned to data points that are difficult to predict based on the other data points whereas low weights are assigned to data points that can be accurately inferred from other data points. To evaluate

our method, a 12-parameter model describing the G1/S transition was examined with two different sets of experimental data: evenly spaced and unevenly spaced data along the time axis. For evenly spaced measurement data, our method demonstrated superiority in estimating parameter set which fit the experimental data very well especially in the dynamically changing region. For the unevenly spaced measurement data set, the dynamically changing region was densely sampled, and the data in the flat region was sparsely sampled. We observed that in such measurement data set, our method did not demonstrate any better performance compared to equal weight cost function method. This is because this sort of data set was strategically done to avoid large amount of redundant information which made the equal weight cost function just as effective. This strategy is often adopted by biologists when designing time series experiments. Our analysis showed a mathematical perspective of why the biologists' intuition of unevenly spaced time points is effective in time series experiments.

Caveat to our method was that it is based on an assumption that dynamically changing part is more important than the steady state part. For the case when the steady state region is critical to understand the process, the iterative algorithm would be modified to reflect the changing assumption. To do this, we would simply change the weight to reciprocal of the weight. Consequently, data points in steady state region would receive larger weights compared to the data points in dynamically changing region. Also, our method would not work for all biological models such as oscillating model. For example, Lotka-Volterra model [89] only has dynamically changing region which means all data points should be treated equally important. In this case, our method would not provide any advantage compared to the equal weighted cost function method. Furthermore, other methods run into overfitting problem when new data points are introduced. One of the ways to address this problem is by introducing regularization term such as L1 (Lasso) and L2 (ridge) which are added to the cost function [90, 91]. However, our weighted cost function can also prevent overfitting problem by re-calculating the weights (relative importance of data points) whenever new

data points are obtained. Lastly, the limitation of our weighted cost function method is that outliers cannot be identified. Outlier would receive the largest weight using our iterative algorithm because outliers cannot be predicted easily using the other data points. Although outliers occur seldomly in biological experiments, the aim of our work is to explain the general behavior of the system biology.

CHAPTER 4
MODELING A MICROFLUIDIC CELL SORTING DEVICE

4.1 Introduction

Mathematical modeling is very efficient to describe, control, and simulate biological systems. In previous sections, we have shown that how to model several biological processes mathematically. In this section, modeling the microfluidic cell sorting device will be discussed as an application.

Microfluidic cell sorting is one of the methods to separate the cells from a mixed batch of cells. The objective of cell separation is to isolate a desired cell type from a mixed batch of cells for experimentation. The microfluidic cell sorting technique can be classified into two groups: Active sorting and passive sorting [92, 93]. The difference between these two techniques is the use of external sources. For active sorting technique, external fields such as electric or magnetic fields was used to separate cells. To make cells respond to the external field, the process for labeling the cell is needed. During this process, some cells are loss and the cost is relatively high. On the other hand, the passive microfluidic cell sorting methods do not require an externally supplied force to separate cells. It depends only on cell biophysical properties, such as cell stiffness or cell size. Therefore, it is simple to operate, and the cost is low because external sources and labeling step are not required. In this chapter, the passive sorting microfluidic device is used because of its benefits compared to the active sorting device. Since the passive sorting techniques only depend on the channel design and biophysical properties of cells, optimizing the device parameters such as device width, length, and height, is crucial step to achieve accurate cell separation. To simulate and optimize the cell separation procedure, a mathematical model describing cell trajectories according to the cell properties, and device parameters is needed. Once the mathematical

model is developed, before performing the cell separating experiment, we can simulate the experiment using different parameter settings.

Microfluidics is a promising technology for biological inquiries at the single-cell level, such as single-cell gene expression for lineage analysis, microfluidic cell sorting, and signaling dynamics [35]. Also, the microfluidic cell sorting device help us to distinguish the cells based on their stiffness. The stiff cells move up and the soft cells move down along the path of the cells flow. This is a very important property because it can be used to detect the cancer cells, such as breast cancer and leukemic cancer cells, which are typically softer than the normal cells [36]. The study of single-cell biomechanical characteristics, such as elasticity, viscosity, stiffness and adhesion is very interesting application [38, 37, 92, 93, 39, 40, 94].

Using a microfluidic channel decorated with ridges that are diagonal with respect to the flow direction (Fig. 4.1), cells are compressed and translated when passing through the channel, and exhibit different trajectories depending on their biomechanical properties. The trajectories are also affected by the channel design, in terms of the ridge height, angle, and spacing. The microfluidic approach for studying cellular biomechanics is highly cost effective compared to atomic force microscopy, and has high throughput similar to flow cytometry. Ridged microfluidic channels have been used to separate cells based on stiffness, adhesion, and viability. This thesis focused on the microfluidic cell sorting device dividing cells based on their stiffness since this is a very important property to detect cancer cells from the mixed cell batch. For example, breast cancer and leukemic cancer cells are typically softer than the normal cells.

4.2 Method

Our ultimate goal is to develop the ODE model that describes the trajectories of the cell flowing through the device. With a mathematical model, the device parameters such as ridge width, ridge separation and ridge angle can be optimized before preforming experi-

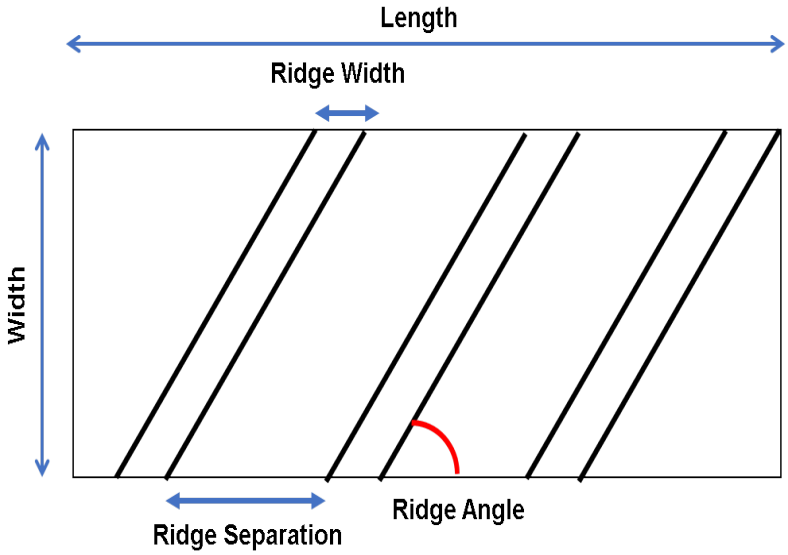


Figure 4.1: **Cartoon illustration of a ridged microfluidic channel** A system that can be used for sorting cells according to their biomechanical properties, e.g. stiffness of cells.

ment. Furthermore, we can simulate different trajectories of cells with different physical parameters setting.

In this work, the ridge induced circulation device (Fig. 4.1) was used in this work. This microfluidic channel is decorated by periodic diagonal ridges. Since these ridges are located in the celling of the device as shown in Fig. 4.1, the cross sectional area under the ridge is smaller than the other arear within the device. Because of the smaller cross sectional area, flowing cells are compressed by the ridges [37, 38, 39, 40]. Therefore, flowing cells are squeezed under the ridges and when they receive pressure from the ridges, they move differently according to their stiffness. Stiff cells receive higher pressure than the softer cells. So, it moves upward compared to the soft cells (The ridges generate circulations that sort cells based on cell properties).

4.2.1 Extracting cell trajectory data

In order to develop mathematical model describing cell trajectories according to different biophysical cell properties, experimentally observed cell trajectory data is needed to fit the model. However, the format of the experimental data is video recording of flowing cells

within the device. Therefore, we need to extract the cell trajectory data from the video.

Experimental data: trajectories of cells

The trajectories of cells contain rich information pertaining to the interactions between the cells and the ridged channel, providing an opportunity for quantifying cell biomechanical properties. By mounting the microfluidic chip on an inverted microscope and a high-speed camera, cells can be recorded when passing through the channel, and this is our observed data. Therefore, the trajectories of cells can be computationally extracted from the recordings. The extracted trajectories will be used for the experimentally observed data when developing the ODE model.

Cell tracking algorithm

Our goal is to automatically extract the trajectories from the gray-scale video recordings [35]. For the example data in Fig. 4.2a, the desired trajectories are shown in Fig. 4.2b. Although time is not shown explicitly, the two cells with entangling trajectories passed through the channel together, and they collided and detached a few times. One cell caught up and collided with the other, the doublet rotated, and the follower became the leader when the two cells detached. This happened again later and the order of the two cells switched back. This example highlights one challenge in this analysis, how to automatically handle the collision and detachment of cells. Our computational pipeline contains three steps: frame-by-frame foreground identification and segmentation, forward matching between consecutive frames, and backward matching between consecutive frames. Since the microfluidic device and the camera are both fixed, the background stays relatively constant, without any rotation, translation or deformation. The baseline intensity of the background varies, due to slight changes in the illumination condition during the experiments. Therefore, the background of each frame can be estimated by the median of nearby frames.

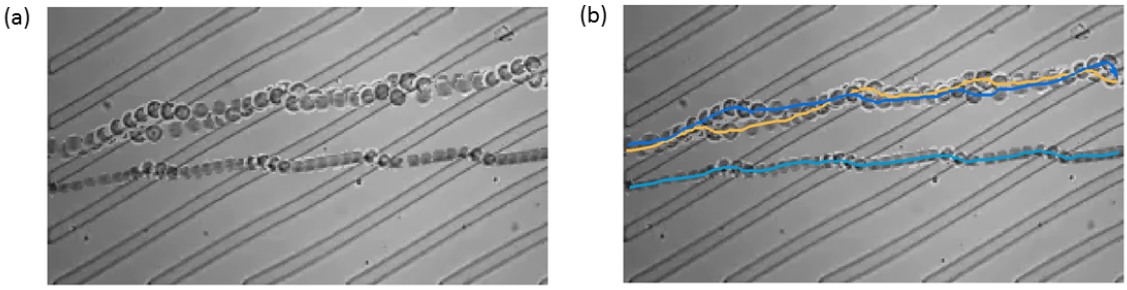


Figure 4.2: **Example data.** (a) a short segment of video recording shown by overlapping multiple frames, (b) desired single-cell trajectories to be extracted.

Foreground identification and segmentation

To identify the foreground objects in an image frame, we first estimate the background by taking the median of nearby frames within a small window, as shown in the illustrative example in Fig. 4.3. A linear regression is performed to predict the image frame by the estimated background. The prediction residues for all the pixels are fitted by a Gaussian distribution. The foreground pixels can be identified by the collection of pixels with residues larger than three standard deviation away from the mean of the residues of all pixels. The foreground is further refined by median filtering to remove noise, and filling in the holes to recover low contrast pixels in the cell nucleus. The foreground pixels can be visualized as a binary image in the bottom-right of Fig. 4.3. We then perform segmentation on the binary representation of the foreground, by computing its connected components. We call each component an "event" in the image frame. This may correspond to a cell, an aggregate of multiple cells, a piece of debris, or noise. In this particular example in Fig. 4.3, the foreground consists of only one event, which is an aggregate of two cells. The outputs of this preprocessing step in the pipeline are the events extracted from each frame. For each event, we compute the (x, y) position of its center, the number of pixels, the radius defined by the maximum distance from the pixels to the center, the mean and standard deviation of the pixel intensities. The cell ID of each event is initialized as NaN, meaning that these events are not associated with any cells yet. The subsequent forward and backward

matching steps will segment the aggregates and associate these events to cells.

The appropriate window size for estimating background depends on the speed of the slow moving cells. If the window size is so small that a slow moving cell appears to be stagnant, it will not be identified as the foreground. In this work, the window size is 300, which corresponds to 0.1 seconds in real time. If a cell is temporarily stuck in the microfluidic channel, moving extremely slowly for > 150 consecutive frames which is half of the window sizethis algorithm can produce incorrect foreground. In addition, the median filtering window size is 5-by-5, which is effective in removing noise in foreground identification. It also removes events smaller than 4 pixels in size, which are typically debris and not of interest.

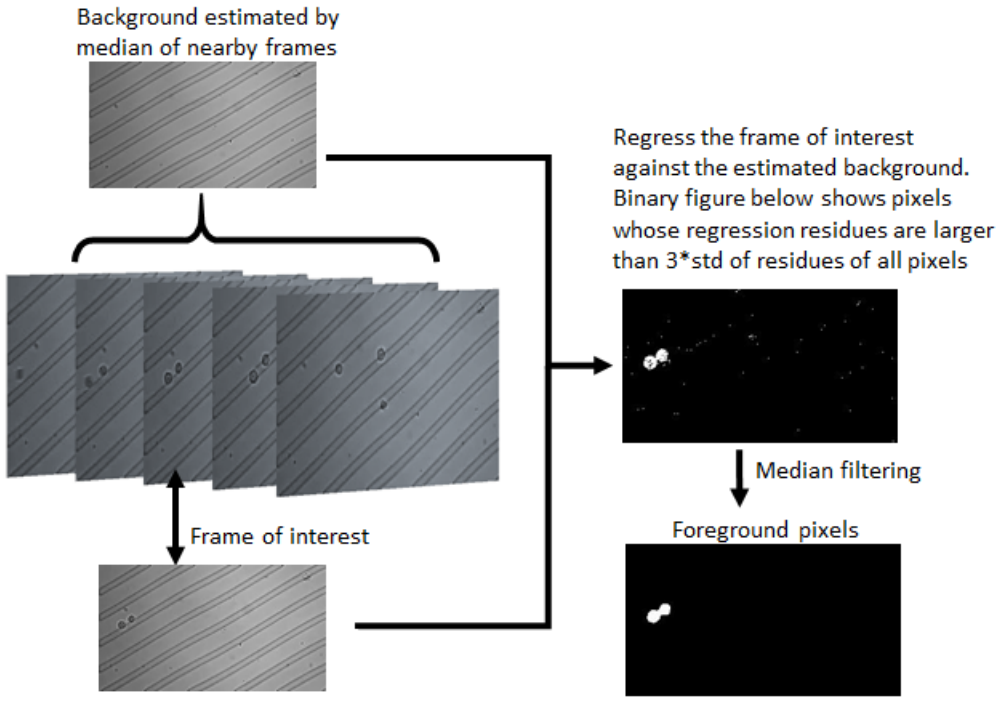


Figure 4.3: **Foreground identification.** Background is estimated by the median of nearby frames. Then, we perform linear regression of the frame of interest against the estimated background, threshold the regression residue to identify foreground pixels. For the last step, we finally perform median filtering to refine the foreground.

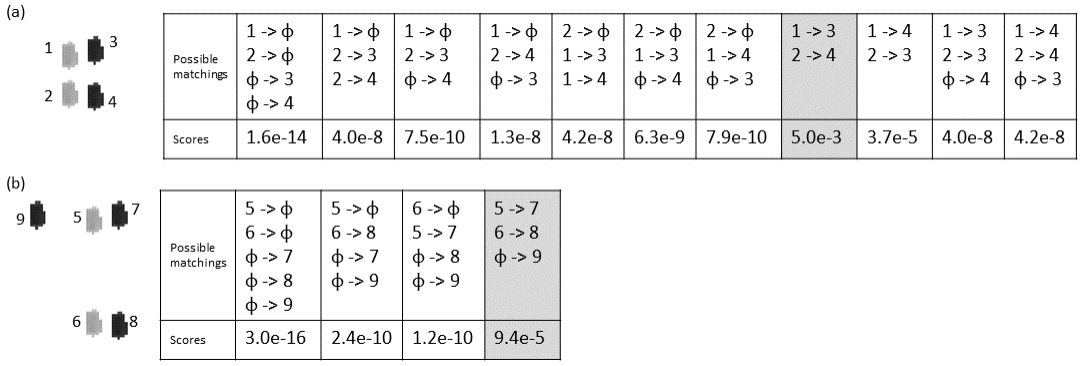
Forward matching of consecutive frames

Each cell passing through the microfluidic channel should appear in a series of consecutive frames. Also, it should ideally produce one event in each of those frames. This may not be true due to various reasons. First of all, a cell with low contrast in one frame may not be detected as an event in the frame-by-frame foreground identification step; a cell can generate multiple events in one frame if part of it is of low contrast which leads to over segmentation; an aggregate of multiple cells in one frame only produces one event. We develop a forward matching algorithm to compare an image frame to its immediate subsequent frame, associate the events to cells, and handle the above situations by merging or segmenting the events when necessary. The algorithm generates all possible matchings of events between the two consecutive frames, scores the possible matchings, and applies the one with the highest score.

(1) Generate all possible sets of matchings of events between the two frames. From the current frame to the next one, each event in the current frame can either disappear, or match to a nearby event in the next frame whose position is further along the flow direction. Similarly, each event in next frame can either suddenly appear, or match to a nearby event upstream of the flow direction in the current frame. Events appear or disappear when cells enter or exit the channel. Two events are considered "nearby" if the distance between their centers is within 5 times of the maximum of their radii. This parameter defines the speed limit of cells that can be tracked. For an extremely fast-moving cell whose center position is far away in the current and next frames, it will be considered as two cells, one disappearing after the current frame and another appearing in the next frame. However, our data does not contain such fast-moving cells, which is guaranteed by the sampling rate of the high-speed camera and the flow rate in the microfluidic channel in our experimental setup.

The ϕ denotes an empty set here. A ϕ-to-one matching represents an event that suddenly appears in the next frame and does not match to any event in the current frame. A one-to-ϕ matching represents an event in the current frame disappears in the next frame and does not

match to any event in the next frame. In addition to ϕ-to-one and one-to-ϕ, the matchings can also be one-to-one, one-to-multiple or multiple-to-one. A one-to-multiple matching represents aggregate becoming detached cells, and a multiple-to-one matching represents multiple cells colliding and forming an aggregate. We do not consider matchings that are multiple-to-multiple. Fig. 4.4 shows two examples. In Fig. 4.4a, compared to events 1 and 2 in the current frame, events 3 and 4 in the next frame are nearby and further downstream the flow direction, left to right. Therefore, both events in the next frame can potentially match to both events in the current frame, allowing a total of eleven possible sets of matchings. In Fig. 4.4b, the total number of possible sets of matchings is four, smaller than the previous case, because not all events in the next frames are downstream and nearby all events in the current frame. Event 9 in the next frame is upstream of all events in the current frame, meaning that it must have just appeared. Although event 8 in the next frame is downstream of both 5 and 6 in the current frame, it can only match to event 6 because it is far away from event 5.



Possible matchings	1 -> ϕ 2 -> ϕ ϕ -> 3 ϕ -> 4	1 -> ϕ 2 -> 3 2 -> 4	1 -> ϕ 2 -> 3 ϕ -> 4	1 -> ϕ 2 -> 4 ϕ -> 3	2 -> ϕ 1 -> 3 1 -> 4	2 -> ϕ 1 -> 3 ϕ -> 4	2 -> ϕ 1 -> 4 ϕ -> 3	1 -> 3 2 -> 4	1 -> 4 2 -> 3	1 -> 3 2 -> 3 ϕ -> 4	1 -> 4 2 -> 4 ϕ -> 3
Scores	1.6e-14	4.0e-8	7.5e-10	1.3e-8	4.2e-8	6.3e-9	7.9e-10	5.0e-3	3.7e-5	4.0e-8	4.2e-8

Possible matchings	5 -> ϕ 6 -> ϕ ϕ -> 7 ϕ -> 8 ϕ -> 9	5 -> ϕ 6 -> 8 ϕ -> 7 ϕ -> 9	6 -> ϕ 5 -> 7 ϕ -> 8 ϕ -> 9	5 -> 7 6 -> 8 ϕ -> 9
Scores	3.0e-16	2.4e-10	1.2e-10	9.4e-5

Figure 4.4: **Matching events in consecutive frames.** Two examples of sets of possible matchings between events in current frame (light gray) and events in the next frame (dark gray), along with scores of the matchings. Numbers are used to label the events.

(2) Score the possible sets of matchings. Cells do not divide or significantly change their shape when moving through the channel. Therefore, their center positions, sizes and pixel intensities should be similar for two events that correspond to the same cell in two consecutive frames. To score matching events, we use differences in these aspects.

The M denotes a particular set of matchings, which may contain one-to-ϕ, ϕ-to-one, one-to-one, one-to-multiple, and multiple-to-one matchings. For an event i, denote its size by P_i, its center position (x_i, y_i), mean and standard deviation of pixel intensities μ_i and σ_i. For a collection of multiple events $j_1, ...j_k$, denote their overall mean and standard deviation of pixel intensities as $\mu_{(j_1,...,j_k)}$ and $\sigma_{(j_1,...,j_k)}$. The basic idea here is to compare the matched events, and penalize changes in size, distance of movement, and differences in pixel intensity distribution. For each individual matching in this set, we score each type of matching as follows.

$$S(i \to \phi) = \left(\frac{1}{P_i}\right)^2 \tag{4.1}$$

$$S(\phi \to j) = \left(\frac{1}{P_j}\right)^2 \tag{4.2}$$

$$S(i \to j) = \left(\frac{1}{P_i - P_j}\right)^2 \left(\frac{1}{|x_i - x_j||y_i - y_j|}\right) \left(1 - \frac{\mu_i - \mu_j}{\max(\mu_i, \mu_j)}\right) \left(1 - \frac{\sigma_i - \sigma_j}{\max(\sigma_i, \sigma_j)}\right) \tag{4.3}$$

$$\begin{aligned} S(i \to j_1, ..., i \to j_k) = & \left(\frac{1}{P_i - \sum_k P_{j_k}}\right)^2 \left(\frac{1}{\max_k(|x_i - x_{j_k}|)\max_k(|y_i - y_{j_k}|)}\right) ... \\ & ... \left(1 - \frac{\mu_i - \mu_{(j_1,...,j_k)}}{\max(\mu_i, \mu_{(j_1,...,j_k)})}\right) \left(1 - \frac{\sigma_i - \sigma_{(j_1,...,j_k)}}{\max(\sigma_i, \sigma_{(j_1,...,j_k)})}\right) \end{aligned} \tag{4.4}$$

$$\begin{aligned} S(i_1 \to j, ..., i_k \to j) = & \left(\frac{1}{\sum_k P_{i_k} - P_j}\right)^2 \left(\frac{1}{\max_k(|x_{i_k} - x_j|)\max_k(|y_{i_k} - y_j|)}\right) ... \\ & ... \left(1 - \frac{\mu_{(i_1,...,i_k)} - \mu_j}{\max(\mu_{(i_1,...,i_k)}, \mu_j)}\right) \left(1 - \frac{\sigma_{(i_1,...,i_k)} - \sigma_j}{\max(\sigma_{(i_1,...,i_k)}, \sigma_j)}\right) \end{aligned} \tag{4.5}$$

The illustrative examples shown in Fig 4.4 are constructed by replicating one cell extracted from our data. All events in these two examples are exactly 53 pixels in size, and share the same pixel intensity distribution. Distances between nearby events range from 10 to 15 pixels. Fig 4.4 shows the overall scores of all possible sets of matchings. This shows us that the most reasonable set receives the highest score in both examples.

(3) Apply the best set of matchings. An algorithm is developed to perform forward matching of consecutive frames, associating events to cells with the scoring function to identify the best set of matchings between events in two consecutive frames. To initialize the algorithm, each event in the first frame is considered as a different cell, and assigned with a

unique cell ID. After that, based on the best set of matchings between events in the first and second frames, the algorithm determines which events in the second frame are associated to cells in the first frame, and which events in the second frame are new cells that should be assigned with new cell IDs. Iteratively, the algorithm examines the second and third frames, the third and fourth frames, and continues until the last two frames. We apply the matchings using the algorithm detailed in Table 4.1 with given cell IDs of events in frame f and the best set of matchings between frames f and $f+1$.

For one-to-ϕ matchings that represent events disappearing, nothing needs to be done. For ϕ-to-one matchings of events appearing, the newly appeared events in frame $f+1$ are assigned with new cell IDs. For a one-to-one matching, the event in frame $f+1$ is associated to the cell ID of the matching event in frame f. For a multiple-to-one matching, the cell IDs of the multiple events in frame f are examined. For each pair of these cells, we identify all previous frames where they co-exist, compute the distance between their centers in each of those frames, and compute the maximum distance. Each pair of cells in the "multiple" forms a pairwise matrix of maximum distances, which represents evidence of whether these "multiple" events in frame f have been separated in previous frames. Since these events in frame f match to one event in frame $f+1$, we define a threshold using the radius of the "one" event. If the smallest element of this pairwise matrix is smaller than the threshold, meaning the two corresponding events have never shown decent separation in the frames $1 \sim f$ analyzed so far, we merge them into one event and assign to it a new cell ID. We then re-compute the pairwise matrix for the remaining events of the "multiple", and check whether any pair should be merged. This is essentially an agglomerative clustering process that merges events in the "multiple" that have never been decently separated. If all the events in the "multiple" are merged together, the matching reduces to the one-to-one case. If not, it is still a multiple-to-one situation. We use the multiple events in frame f to construct templates to segment the matched one event in frame $f+1$. The templates are constructed by shrinking the distances among the multiple events by a varying scalar

($s <= 1$) and rotating them by a varying angle ($\theta \in [-\pi, \pi)$). Using the template that overlaps most with the "one" event, we segment the "one" event into multiple events, each of which is assigned with the cell ID of the corresponding event in frame f. Fig 4.5 shows an example of how a three-to-one matching is performed. For a one-to-multiple matching, we also first examine the pairwise distances between the multiple events. Since they have not yet been associated to any cells, their pairwise distances can only be computed based on the frame $f + 1$ which they belong to. Using the radius of the "one" event in frame f as threshold, we perform the same agglomerative merging as in the multiple-to-one situation above. If all the events in the "multiple" are merged together, the matching reduces to one-to-one. If not, the multiple events after merging are considered as newly appeared cells and assigned with new unique cell IDs.

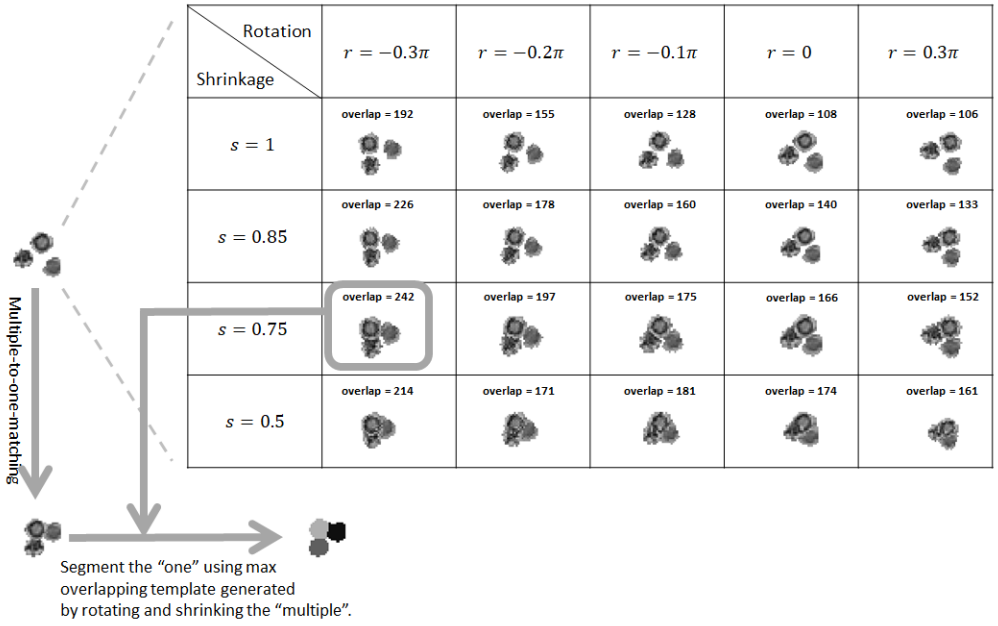


Figure 4.5: **Multiple-to-one matching.** By shrinking the distances among the multiple events and rotating them together, templates are generated to represent possible configurations of the multiple in the next frame. Templates are evaluated by their number of pixels that overlap with the one event in the next frame. The maximal overlapping template is used to segment the one event in the next frame into multiple pieces, turning the multiple-to-one matching into one-to-one.

Table 4.1: Algorithm for forward matching of consecutive frames.

```
Each event in the first frame is assigned a unique cell ID.
for f = 1 : number of frames - 1 do
    Compare the events from frame f to frame f + 1, generate all possible sets of matchings and score them.
    Identify the best set of matching with highest score.
    for each matching in the best set do
        if matching represent event disappearing (i → φ) then
            Do nothing.
        end if
        if matching represent event appearing (φ → j) then
            Assign a new cell ID to event j.
        end if
        if one-to-one matching (i → j) then
            Assign the cell ID of event i to event j.
        end if
        if multiple-to-one matching (i_1 → j, ..., i_k → j) then
            Compute pairwise distance for i_1, ..., i_k based on the associated cells in frames 1 ∼ f.
            while (The smallest pairwise distance < radius of event j) do
                Merge the two corresponding events, and assign a new cell ID to the merged event
                Recompute the pairwise distance in based on the corresponding cells in frames 1 ∼ f.
            end while
            if (Events i_1, ..., i_k are merged to one event i) then
                Assign the cell ID of event i to merged event j.
            else
                Use the remaining events in frame f as templates to split event j into multiple events.
                Assign the cell ID of each remaining event in frame f to the events generated by splitting j.
            end if
        end if
        if one-to-multiple matching (i → j_1, ..., i → j_k) then
            Compute pairwise distance for events j_1, ..., j_k in frame f + 1.
            while (The smallest pairwise distance ¡ radius of event i) do
                Merge the two corresponding events.
                Recompute pairwise distance.
            end while
            if (Events j_1, ..., j_k are merged to one event j) then
                Assign the cell ID of event i to merged event j.
            else
                Assign new and unique cell IDs to each remaining event after the merge.
            end if
        end if
    end for
end for
```

Backward matching of consecutive frames

When events are merged by agglomerative clustering in the multiple-to-one or one-to-multiple matchings during the forward matching, the merged events are assigned with new cell IDs. When there is a one-to-multiple matching, the multiple events in the next frame receive new cell IDs. Those events with new cell IDs are not associated to any events in the previous frames, but their corresponding cells may exist in the previous frames. The backward matching addresses this issue.

The backward matching algorithm is almost identical to the forward matching algorithm, except for two key differences. First of all, the backward matching process starts from the last frame. Then, it goes back in time to match events in the current frame to events in its previous frame. Second, the pairwise matrix of maximum distances between the multiple events is computed based on their associated cells in all frames when the agglomerative merging in multiple-to-one or one-to-multiple matchings is performed. The combination of forward and backward matching enables accurate tracking of cells involved in complex sequences of collisions and detachments. We assume that we have a short video of four frames, and first row of Fig 4.6 represents the events obtained from foreground identification and segmentation. The numbers of events in these four frames are two, one, two, and three. Upon initialization of the algorithm, the two events in the first frame receive unique cell IDs (1, 2). Each subsequent row shows the tracking result after one step of the forward and backward matching. From the first to the second frame, the best matching is two-to-one. Since the two events in the first frame are not well separated compared to the size of the one event in the second frame, they are merged into one event and assigned with a new cell ID (3). After the first iteration of forward matching, there is only one event in the first frame. The matching from the second to the third frame is one-to-two, and the two events in the third frame receive new cell IDs (4, 5) because they are decently separated compared to the size of the one event in the second frame. Similarly, the matching from the third and fourth frame is also one-to-two with decent separation between the two, and therefore, new cell IDs (6, 7) are assigned to events in the fourth frame. The backward matching from the fourth to third frame has a two-to-one matching. Since the two (6, 7) are decently separated, the one (5) is split into two events and assigned with the corresponding cell IDs. The matching from the third and second frame is three-to-one. Since the three events (cells 4, 6, 7) have been well separated in other frames, the one (3) is split to three events and assigned with appropriate cell IDs. The last step of backward matching is also three-to-one, and follows the same operation. At the end of the forward and backward

matching process, all frames have three events which are correctly associated to three cell IDs. Performance of the matching algorithm can be affected by the quality and availability of the data. In the above example, if the fourth frame does not exist, the algorithm will only be able to identify two cells, one is the upper cell (4), and the other is the doublet (5) of the middle and bottom cells in the third frame.

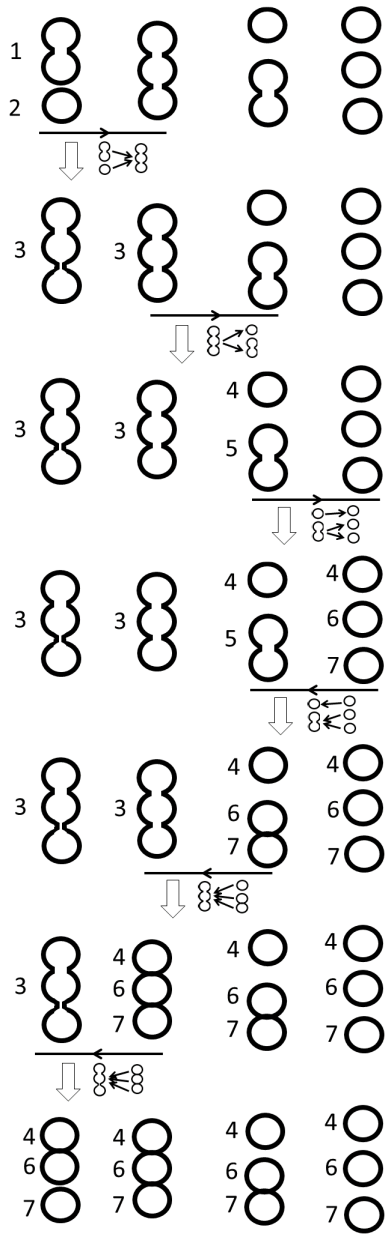


Figure 4.6: **The forward and backward matching process.** Each column corresponds to one image frame. Left-to-right is the forward directions. Numbers are used to indicate the cell ID associated to each event. Each vertical arrow represents one step of the forward or backward matching between two consecutive frames. As the algorithm proceeds, multiple-to-one matchings either cause the multiple merge or the one to split. One-to-multiple matchings either cause the multiple to merge (not contained in this example), or cause the multiple to receive new cell IDs.

4.2.2 Modeling cell trajectory in ridged microfluidic device

As a first step for establishing mathematical model for cell trajectories, the velocity field of water within the microfluidic channel was obtained from the COMSOL multiphysics software. To simulate the flowing cell trajectory, all forces exerted to the cell should be considered. Then we can use the Newtons law, force is mass times acceleration. The acceleration is the first derivative of velocity with respect to the time. Velocity is the first derivative of position with respect to the time. Therefore, If we know the force exerted to the cell, we can get the position of the cell, which is the cell trajectory (Eq. 4.6).

$$\begin{aligned} \tfrac{d}{dt} Position &= Velocity \\ \tfrac{d}{dt} Velocity &= Acceleration \end{aligned} \tag{4.6}$$

Basically, the drag force should be considered (Eq. 4.7), where μ_w is the viscosity of the water, r is the radius of a cell, and V_{water}, V_{cell} represent the velocity of water and a cell, respectively. The drag force is a force acting to opposite the relative motion of moving object with respect to a surrounding fluid (Vp-Vw). Therefore, the direction of drag force is the same with the cell flowing so, it propels cells forward.

$$F_{drag}(t) = 3\pi\mu_w r(V_{water} - V_{cell}) \tag{4.7}$$

When flowing cells confront periodic ridges in a microfluidic channel, they experience elastic force and frictional force. When the cell is entering the ridge or leaving the ridge, some part of the cell is deformed and because of the deformation, the cell experiences the force perpendicular to the ridge, shown in Fig 4.7. When the cell is leaving the ridge, some part of the cell is escaped the ridge and it experiences the force perpendicular to the ridge. The Hertzian contact stress (Eq. 4.8) is usually used to refer the stress close to the area of contact between two spheres of different radius [95]. In this case, I assumed the radius of

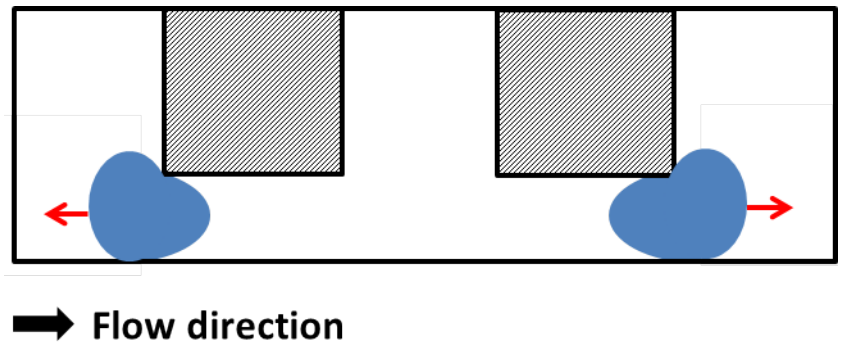


Figure 4.7: **Cartoon illustration of elastic force exerted to a flowing cell under the ridge** When the flowing cell is entering or leaving the ridge, the elastic force (red arrow) is exerted to the cell. The direction of the force is perpendicular to the ridge.

the ridge is infinite. In Eq. 4.8, E is Young's modulus, u is Poisson ratio of a cell. Youngs modulus (E) represents the cell stiffness, and its definition is the ratio between the applied pressure and the strain. So, when we press the cells, there is strain because cell is elastic. If this Youngs modulus is big, then the cell is relatively stiff, if this modulus is small, then the cell is relatively soft.

$$F_{elastic}(t) = \frac{4}{3}\frac{E}{1-u^2}\sqrt{r}strain^{1.5} \tag{4.8}$$

The magnitude of the force is proportional to the deformed area of the cell, which is the difference between cell diameter and ridge gap.

The frictional force direction is opposite to the moving direction, and its magnitude is proportional to the normal force,N, and frictional coefficient ,μ_{fric} (Eq. 4.9).

$$F_{fric}(t) = \mu_{fric}N \tag{4.9}$$

We assumed that the normal force is the z-direction component of the elastic force.

If the cell is soft, the compression force is weak and then cells move with fluid flow streamlines. And stiff cell experiences strong deformation forces and moves upward. Therefore, soft and stiff cells migrate to opposite sides of the ridged microfluidic chan-

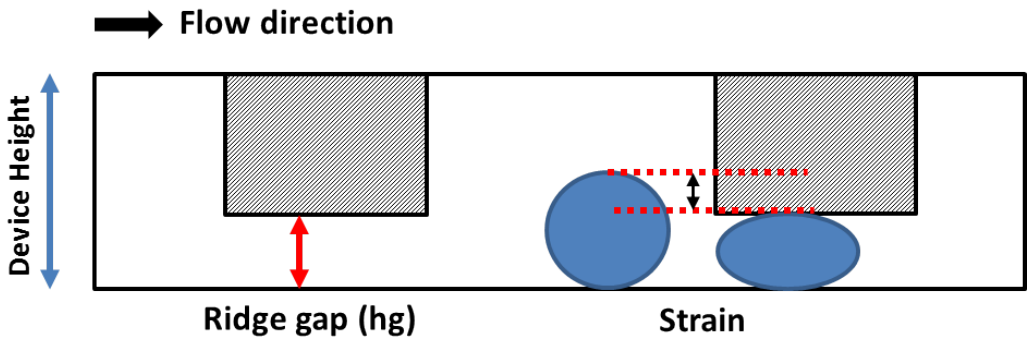


Figure 4.8: Cartoon illustration of strain of a flowing cell under the ridge. When the flowing cell is deformed due to the ridge, the strain (ϵ) can be calculated by difference between diameter of a cell and ridge height.

nel, separating according to their mechanical stiffness.

4.3 Results

4.3.1 Experimental data: trajectories of cells

The recordings under 0CD condition contained 23312 frames and 23416 events in total, indicating that the 0CD recordings were sparse, and each frame contained only one cell on average. In fact, 36% of the frames were empty, 38% of the frames contained one cell, and 24% of the frames contained 2 or more cells. This was because K562 cells under 0CD condition were stiff and easily slowed down by the ridges, and thus, they tended to clog the microfluidic channel if the cell concentration was higher and more cells were passing through the channel simultaneously. The cell concentration under the 1.5CD condition was higher. The recordings under 1.5CD condition contained 9718 frames with 27026 events, translating to roughly 3 events per frame.

The forward and backward tracking process associated the events to cells and trajectories. Trajectories that started and ended at the boundaries of the field of view were considered as correctly tracked trajectories, whereas incorrectly tracked trajectories were those either started or ended in the middle of the field of view. Under the two perturbation conditions, the percentage of events associated to correctly tracked trajectories were

92% and 97% respectively. Fig. 4.9 shows that events associated to incorrectly tracked trajectories were much smaller compared to the correctly tracked cells. Those incorrectly tracked events were typically small debris that had low contrast and high speed when moving through the device.

Fig. 4.10 shows overlays of the correctly tracked trajectories. We can see that cells under 0CD condition drifted up along the y-direction, whereas cells under 1.5CD condition exhibited a slightly negative drift along the y-direction. This was a significant difference as shown in Fig. 4.11a.

For each cell/trajectory, we computed its average speed when overcoming the ridges, and its average speed when traveling in the gaps between the ridges. The results were visualized in Fig. 4.11b. Cells under 1.5CD condition scatter close to the 45 degree line, meaning that their speed on ridges and speed in gaps are similar, because they were soft and could easily deform and overcome the ridges. The variation of speed was tightly correlated with size, with smaller cells traveling faster and large ones traveling slower. Under 0CD condition, cells were stiff and more affected by the ridges, and thus, their speed on ridges was much slower than their speed in gaps. The two populations in Fig. 4.11b can be well-fitted by two straight lines, suggesting that the stiffness of cells was relatively constant within each population, and the slopes were useful for quantifying the stiffness.

The trajectories enabled quantification of subtle changes of cells as they overcame several ridges. For each trajectory, we computed the cell's average speed when passing each ridge, and normalized by its average speed on the first ridge. For cells under 0CD condition, the mean and standard deviation of the normalized average speed per ridge was shown in Fig. 4.11c. We observed that stiff cells tended to travel slightly faster as they passed more ridges. Given the large standard deviation shown in Fig. 4.11c, this trend of increasing speed was not statistically significant. However, it was consistent with previous observations that stiff cells can become softer after repeated biomechanical perturbation. In contrast, Fig. 4.11d showed that the soft cells maintained constant speed with respect to

the number of ridges they encountered.

(a)

	0CD	1.5CD
Total number of frames	23312	9718
Total number of events	23416	27026
Correctly tracked events	21560	26147
Percentage of correctly tracked events	92.07%	96.75%

(b)

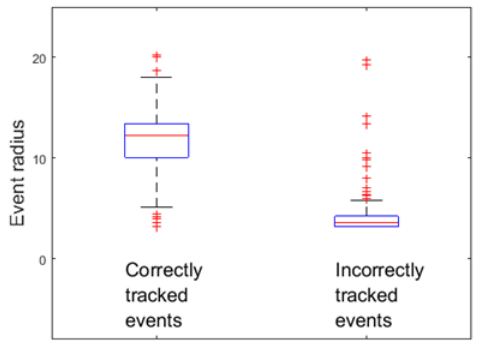


Figure 4.9: **Summary of tracking results on recordings of cells under two perturbation conditions.** (a) Size of the data and tracking performance. (b) Comparing the size of events associated to correct and incorrect trajectories, the incorrectly tracked events are mostly small debris.

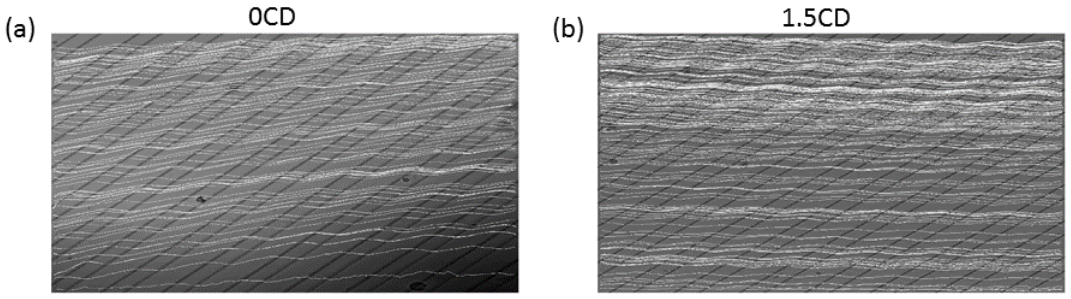


Figure 4.10: **Visualization of tracking results.** Overlay of the correctly tracked trajectories on the background of the recordings.

4.3.2 Modeling cell trajectory in ridged microfluidic device

Velocity field of water: COMSOL

In order to consider forces exerted to the cells in the microfluidic device, the velocity of the water surrounding the moving cell is required. Once specification of the microfluidic device is determined, velocity of flowing water can be simulated through the COMSOL software, and the velocity field data (water) can be extracted. The device parameters was defined as follows: Height = 22μm, length = 1400μm, width = 560μm, gap height = 8μm,

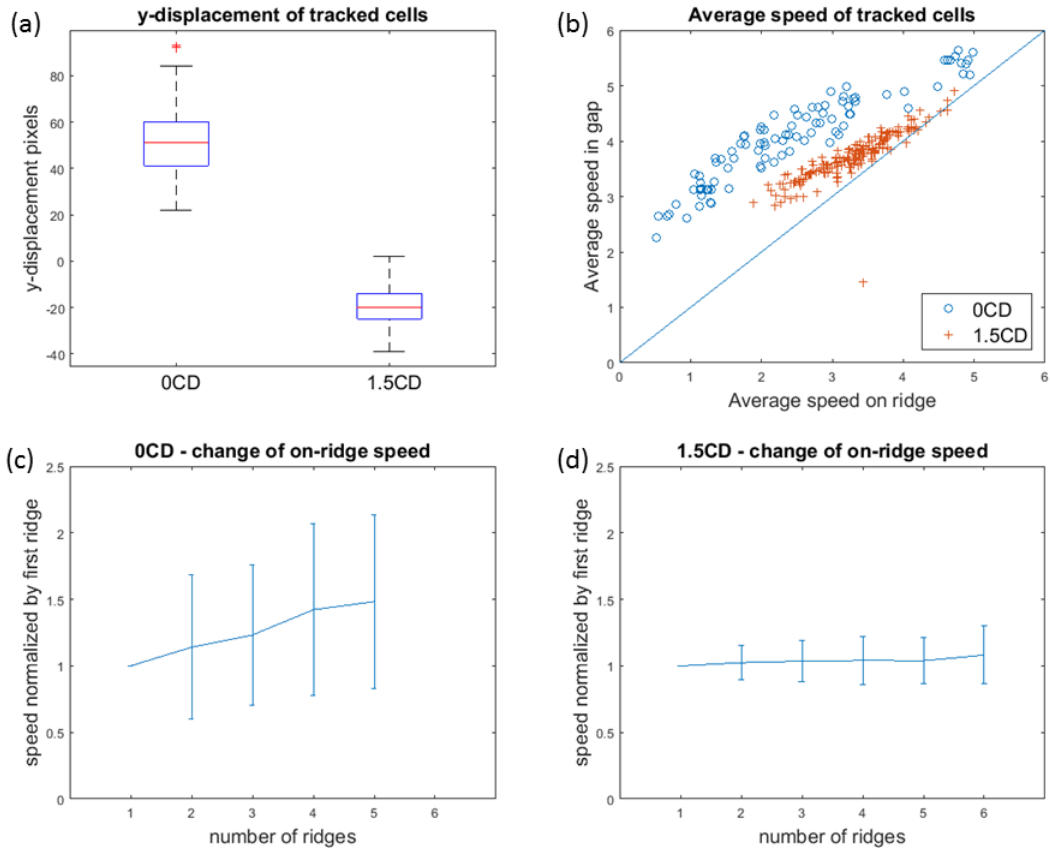


Figure 4.11: **Comparison of 0CD (stiff) and 1.5CD (soft) cells.** (a) Stiff cells drift up along the y-direction, whereas soft cells tend to have a slightly negative drift. (b) The speed on ridge of stiff cells is smaller than their speed in gap. Soft cells are less affected by the ridges. (c) Stiff cells tend to travel faster after passing each ridge, whereas (d) soft cells travel at a relatively constant speed irrespective of the ridges.

and ridge angle $=\frac{\pi}{4}$. Fig. 4.12 shows that the water velocity at 5μm device height within the device. As shown in the figure, since the water flows from the left to right direction within the device, the velocity of water in X direction is always positive. Also, the magnitude of the velocity in X direction is increase under the ridges because the cross sectional area is decrease. The Y direction of the water velocity becomes negative under the ridges. For the Z direction of the water velocity, the direction of velocity is negative when the water entering the ridges, while the direction becomes positive when they escape from the ridge.

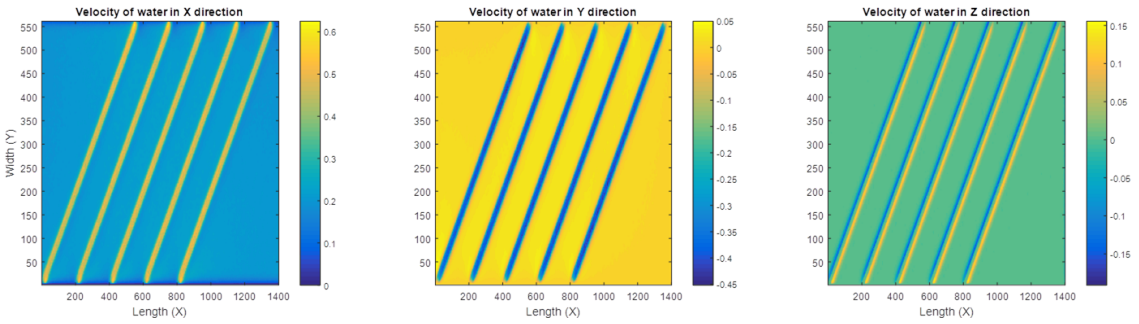


Figure 4.12: **Velocity field of water within the ridged microfluidic device at 5μm height** To visualize the water flow within the microfluidic device, velocities of flowing water in X,Y, and Z direction are shown. X,Y, and Z represent direction of length, width, and height of the microfluidic device. Water flows from the left to right direction. Color represents the water velocity (m/s).

Simulation results: cell trajectory

Based on the velocity field of water within the microfluidic device, we can simulate the cell trajectory by using the method discussed in the Method section. Fig. 4.13 shows simulated cell trajectory of 4μm diameter cell. As shown in this figure, the 4μm diameter cell is smaller than the gap size (8μm), therefore this cell is not compressed under the ridges. As a result, this cell translated when passing through the channel without experiencing the elastic force and frictional force, so its trajectory is almost same with the water streamline. This simulation result is similar to the fig. 4.10(b), which is a collection of trajectories of soft cells. From this comparison, we can imagine that if a cell is bigger than the ridge gap and very soft which means it has very low Young's modulus, then it will receive very small amount of elastic and frictional force. This is because these forces are proportional to Young's modulus.

As shown in Fig. 4.14, if the trajectory of the cell is visualized in 3-D, a helical pattern is shown. This helical pattern is induced by periodic ridges.

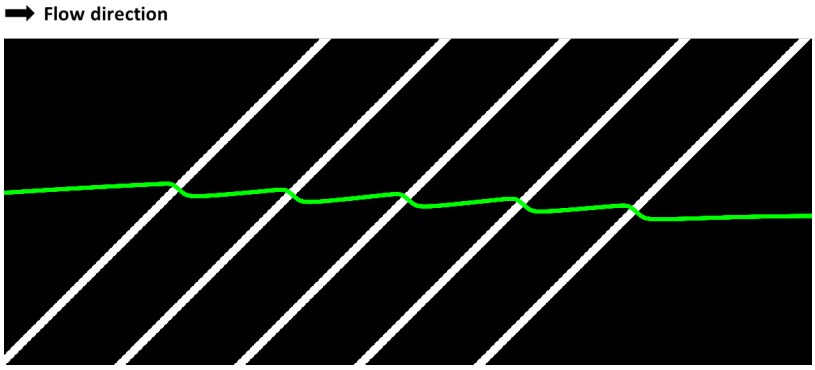


Figure 4.13: **Simulated cell trajectory** The white pixels represent diagonal ridges and blue curve represents the cell trajectory. The green trajectory represents the trajectory of 4μm cell.

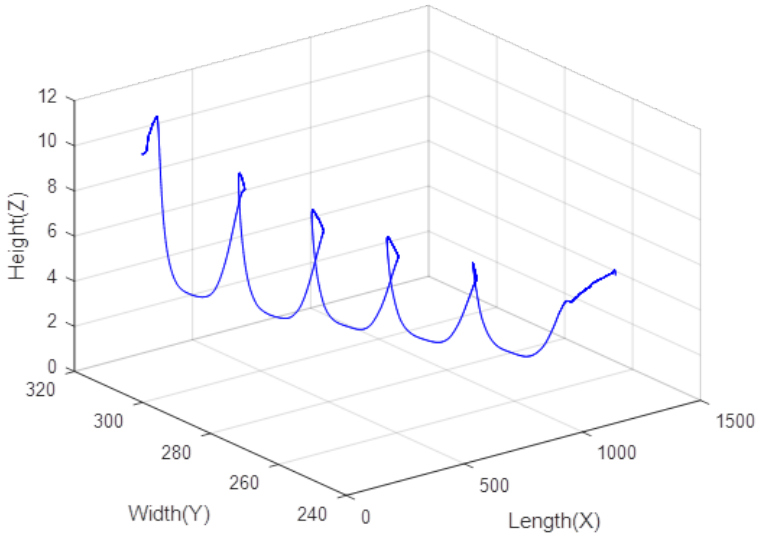


Figure 4.14: **Simulated cell trajectory in 3-D** Trajectory of 4μm cell has a helical pattern.

4.4 Conclusion and Discussion

To develop the ODE model describing cell trajectory within the ridged microfluidic cell sorting device, the experimental data, cell trajectories, was needed. To obtain the cell trajectory, a computational algorithm was developed. This algorithm can automatically and accurately extract the cell trajectories in high-speed video recordings of microfluidic cell sorting devices. We tested the algorithm on recordings of K562 cells under two perturbation conditions, representing stiff and soft cells. The algorithm successfully handled the collision and detachment of cells. We showed that the automatically extracted trajectories correctly captured the difference in stiffness between the two perturbation conditions. The tracked trajectories revealed the subtle increase in speed when the stiff cells pass through consecutive ridges, which is an indication that cell biomechanical properties may change when passing through the ridges. The accurately tracked trajectories enables future efforts of optimizing device design, for the purposes of modeling and quantification of the dynamical changes of cell biomechanical properties in the context of microfluidics.

To construct the ODE model representing cell trajectory depending on different biophysical properties of cells, we have discussed three forces exerted to the cell when they are flowing within the device: Drag force, Elastic force, and Frictional force .There are still rooms for improvement in the model but this is still an on-going project which will be continued by the next graduate student.

CHAPTER 5
CONCLUSION AND FUTUREWORK

Dynamic behaviors of biological processes can be described in Ordinary Differential Equations (ODEs). By describing biological processes in ODEs, we can control and predict the behaviors of biological processes. The ODE-based model contains several unknown model parameters such as reaction rates and they should be estimated based on experimentally observed data. The problem is that the amount of experimental data is almost always limited and the model complexity is relatively high. This imbalance between the insufficient data and high model complexity makes the parameter estimation more challenging. To address this problem, experimental design method and model reduction method have been developed. The most informative experiment can be selected among other candidate experiments using experimental design method. By performing the selected experiment, the number of data is increase, and as a result, parameter estimation can be improved. Model reduction method finds the insensitive parameters, and by removing or combining these insensitive parameters, complexity of the model can be reduced with maintains its ability to fit the data. Furthermore, model reduction method can be used to find the key part of the system.

Currently, these two methods are considered as two distinct approaches. However, in chapter 2, the possibility of unified framework of these two methods was shown. In order to consider these two methods in a common framework, we applied the geometric concept. We considered the mathematical model as a manifold living in a data space, and considered experimentally observed data point as a point in the data space. Also, parameter estimation can be viewed as projecting the data point onto the model manifold. In this manner, we can perform both experimental design and model reduction by examining the singularity around the projected point on the manifold. We have shown that what experimental design

method does is expanding the model manifold around the projected point. Model reduction method can identify the nearest manifold boundary which suggests appropriate forms of reduced models.

Our new unified framework will lead to a new technique for reducing the information gap between limited data and complex mathematical model efficiently by identifying the most informative experiment and finding the key controlling mechanism of the system at the same time. In addition, this unified approach can bring intuitive geometric interpretation in systems biology. However, the expected limitation of this approach is that as the dimension of the data space increases, the computational cost will increase because finding manifold boundary is not that simple. Also, visualizing the changing model manifold will not be an easy task because of the high dimension of the data space.

In chapter 3, a new novel approach in parameter estimation is introduced. The motivation of our approach stems from the imbalance between high complexity of the model and limited availability of experimental data, which brings about the ill-conditioned parameter estimation. Another motivation of our approach comes from the assumption of the experimental design method. Experimental design method assumes that all experiments contain different amount of information. In the same manner, we assume that data points obtained from one single experiment also contain different amount of information. To apply this assumption to the least square cost function, the cost function was re-formulated by embedding the different amount of information of each data point.

One of the benefits of our approach is that additional experiments are not required to improve parameter estimation. Instead of suggesting new experiments to obtain more data to estimate parameters more accurately, our approach uses a weighted cost function which improves the parameter estimation by giving a different weight for each data point. A different weight for each data point is quantified based on the uncertainty of each data point given the other data points. By giving a different weight for each data point, we have improved the parameter estimation problem. In addition, we have shown that our approach

can reduce the redundancy among the experimentally observed data points. This also shows that the mathematical perspective of why the biologists' intuition of unevenly spaced time points is effective in time series experiments. This ability which is reducing redundancy of data can be utilized to experiment setting, especially when there is not much information about which time points are crucial to explain the behaviors of the system.

Some limitations of our weighted cost function approach are identified. First of all, our method would not work for all biological models such as oscillating model. For example, Lotka-Volterra model [89] only has dynamically changing region which means all data points should be treated equally important. In this case, our method would not provide any advantage compared to the equal weighted cost function method. However, the case that the steady-state region is more important than the dynamically changing region can apply our approach with a simple change in our approach. Since the assumption of our approach is that dynamically changing part is the most important in biological behavior, we need to change this assumption to deal with this different case. For this case steady state region is important compared to other region, the weights can be embedded to the cost function as a reciprocal of the weights. Then, the steady-state region will receive higher weight than the other region. Another limitation of our weighted cost function method is that outliers cannot be identified. Outlier would receive the largest weight using our iterative algorithm because outliers cannot be predicted easily using the other data points. Although outliers occur seldomly in biological experiments, the aim of our work is to explain the general behavior of the system biology. Next, our approach is still not perfect for parameter estimation as many different parameter settings still can fit the data well. One of the future works could be somehow combining model reduction technique and our weighted cost function to further improve the parameter estimation. Lastly, one expected challenging problem is that if our weighted cost function is applied to more complex biological model, the computational cost will increase drastically. Since our approach uses sensitive equations when we calculate the fisher information matrix to quantify the relative

importance of each data point, the number of model parameters will affect to computational cost. Therefore, reducing computational cost can be another future work to improve our approach.

In chapter 4, we discussed the computation approach in modelling the cell flow in ridged microfluidic channel device to demonstrate the procedure and the practicality of modeling the biological system. The ridged microfluidic channel device is used to sort the cell depending on the biophysical properties of cells such as cell stiffness or size. The mathematical model approach describing the microfluidic cell trajectory can help us tremendously and save time by eliminating the unsuccessful experiments. For example, before performing an experiment for sorting the cells, we can simulate the experiment and optimize the device parameters such as length, width and height of the device, and angle of the ridge as well as the cell size and cell stiffness.

To develop the ODE model describing cell trajectory within the ridged microfluidic cell sorting device, the experimental data of cell trajectories was needed. However, we only had the video recordings of cells flowing through microfluidic channel. Therefore, the new cell tracking algorithm was developed and introduced in this chapter. This algorithm automatically and accurately extracted the cell trajectories in high-speed video recordings of microfluidic cell sorting devices. Although the accuracy of the tracking algorithm is almost 92%, there is still room for improvement. For example, the speed of our algorithm can be improved because when the video containing more than 20 cells per frame, the speed of our algorithm slowed down. In addition, we can improve our pre-processing step to obtain more accurate cell information such as cell size. Since the contrast within in one cell is different, for instance, some parts of a cell is dark and some part of a cell is bright, these kinds of cells can give us incorrect size information of cells. Therefore, in order to utilize our tracking algorithm into different projects such as tracking cell size changing within the microfluidic channel, these improvements will be very helpful to track changing information of moving cells accurately.

The next step was to construct the ODE model representing cell trajectory depending on different biophysical properties of cells. We have discussed three forces exerted to the moving cell when they are flowing within the device: Drag force, Elastic force, and Frictional force. The developed model was able to predict the flow of the cells correctly when the cell was smaller than the ridge gap. However, the current model was unable to accurately describe the trajectories when the cell is bigger than the ridge gap. This is where the model needs an improvement and this is still an on-going project to be continued by the next graduate student.

In order to obtain more accurate simulation results, we also need to think about other forces exerted to the moving cells. For example, the elastic force we used is based on the assumption that the cell is completely uniaxial compressed by the ridge. However, the elastic force would be different in the case when the cell is entering or leaving the ridge. As only part of the cell is in contact with the ridge, uniaxial compression cannot be used as elastic force. Thus, in this case, the direction and magnitude elastic force exerted to the cell would be different and subsequently, frictional force would be different. Reflecting this different force information exerted to one cell would be the most challenging part to develop the ODE model for this microfluidic device.

Once the ODE model describing cell flowing within the microfluidic device is developed, our weighted cost function introduced in chapter 3 should be used to identify the important part of the cell trajectory. As the simulation result of cell trajectories would have a lot of redundancy, our weighted cost function approach can make the optimization of the microfluidic channel device more efficiently.

REFERENCES

[1] A. Aderem, "Systems biology: Its practice and challenges," *Cell*, vol. 121, pp. 511–513, 4 2005.

[2] U. Sauer, M. Heinemann, and N. Zamboni, "Getting closer to the whole picture," *Science*, vol. 316, no. 5824, pp. 550–551, 2007.

[3] S. D. Findlay and P. Thagard, "How parts make up wholes," *Frontiers in Physiology*, vol. 3, p. 455, 2012.

[4] A. D. Lander, "A calculus of purpose," *PLOS Biology*, vol. 2, no. 6, Jun. 2004.

[5] E. A. Sobie, Y.-S. Lee, S. L. Jenkins, and R. Iyengar, "Systems biology—biomedical modeling," *Science Signaling*, vol. 4, no. 190, tr2–tr2, 2011.

[6] D. Machado, R. S. Costa, M. Rocha, E. C. Ferreira, B. Tidor, and I. Rocha, "Modeling formalisms in Systems Biology," *AMB Express*, vol. 1, p. 45, 2011.

[7] E. Bartocci and P. Li, "Computational modeling, formal analysis, and tools for systems biology," *PLOS Computational Biology*, vol. 12, no. 1, pp. 1–22, Jan. 2016.

[8] K. Sachs, D. Gifford, T. Jaakkola, P. Sorger, and D. A. Lauffenburger, "Bayesian network approach to cell signaling pathway modeling," *Sci. STKE*, vol. 2002, no. 148, pe38, 2002.

[9] F. Fages, S. Gay, and S. Soliman, "Inferring reaction systems from ordinary differential equations," *Theoretical Computer Science*, vol. 599, no. Supplement C, pp. 64 –78, 2015, Advances in Computational Methods in Systems Biology.

[10] I. Koch, "Petri nets in systems biology," *Software & Systems Modeling*, vol. 14, no. 2, pp. 703–710, 2015.

[11] H. Kitano, "Computational systems biology," *Nature*, vol. 420, no. 6912, pp. 206–210, 2002.

[12] B. B. Aldridge, J. M. Burke, D. A. Lauffenburger, and P. K. Sorger, "Physicochemical modelling of cell signalling pathways," *Nature Cell Biology*, vol. 8, p. 1195, 2006.

[13] J. Anderson, Y.-C. Chang, and A. Papachristodoulou, "Model decomposition and reduction for large-scale networks in systems biology," *Automatica*, vol. 47, no. 6, pp. 1165–1174, Jun. 2011.

[14] T. Quaiser, A. Dittrich, and M. Schaper Fredand Mönnigmann, "A simple work flow for biologically inspired model reduction - application to early jak-stat signaling," *BMC Systems Biology*, vol. 5, no. 1, p. 30, 2011.

[15] A. F. Villaverde, D. Henriques, K. Smallbone, S. Bongard, J. Schmid, D. Cicin-Sain, A. Crombach, J. Saez-Rodriguez, K. Mauch, E. Balsa-Canto, P. Mendes, J. Jaeger, and J. R. Banga, "Biopredyn-bench: A suite of benchmark problems for dynamic modelling in systems biology," *BMC Syst Biol*, vol. 9, 2015.

[16] B. B. Machta, R. Chachra, M. K. Transtrum, and J. P. Sethna, "Parameter space compression underlies emergent theories and predictive models," *Science*, vol. 342, no. 6158, pp. 604–607, 2013.

[17] J. R. Banga and E. Balsa-Canto, "Parameter estimation and optimal experimental design," *Essays In Biochemistry*, vol. 45, pp. 195–210, 2008.

[18] C. Kreutz and J. Timmer, "Systems biology: Experimental design," *FEBS Journal*, vol. 276, no. 4, pp. 923–942, 2009.

[19] P. Meyer, T. Cokelaer, D. Chandran, K. H. Kim, P.-R. Loh, G. Tucker, M. Lipson, B. Berger, C. Kreutz, A. Raue, B. Steiert, J. Timmer, E. Bilal, H. M. Sauro, G. Stolovitzky, and J. Saez-Rodriguez, "Network topology and parameter estimation: From experimental design methods to gene regulatory network kinetics using a community based approach," *BMC Systems Biology*, vol. 8, no. 1, p. 13, 2014.

[20] J. Vanlier, C. A. Tiemann, P. A. J. Hilbers, and N. A. W. van Riel, "A bayesian approach to targeted experiment design," *Bioinformatics*, vol. 28, pp. 1136–42, 2012.

[21] J. Liepe, S. Filippi, M. Komorowski, and M. P. H. Stumpf, "Maximizing the information content of experiments in systems biology," *PLOS Computational Biology*, vol. 9, no. 1, pp. 1–13, Jan. 2013.

[22] X. Huan and Y. M. Marzouk, "Simulation-based optimal bayesian experimental design for nonlinear systems," *Journal of Computational Physics*, vol. 232, no. 1, pp. 288 –317, 2012.

[23] D. Faller, U. Klingmller, and J. Timmer, "Simulation methods for optimal experimental design in systems biology," *SIMULATION*, vol. 79, no. 12, pp. 717–725, 2003.

[24] M. K. Transtrum and P. Qiu, "Optimal experiment selection for parameter estimation in biological differential equation models," *BMC Bioinformatics*, vol. 13, no. 1, p. 181, 2012.

[25] O. Radulescu, A. N. Gorban, A. Zinovyev, and V. Noel, "Reduction of dynamical biochemical reactions networks in computational biology," *Front Genet*, vol. 3, 2012.

[26] R. Krüger and R. Heinrich, "Model reduction and analysis of robustness for the wnt/beta-catenin signal transduction pathway.," *Genome informatics. International Conference on Genome Informatics*, vol. 15 1, pp. 138–48, 2004.

[27] Z. P. Gerdtzen, P. Daoutidis, and W.-S. Hu, "Non-linear reduction for kinetic models of metabolic reaction networks," *Metabolic Engineering*, vol. 6, no. 2, pp. 140 –154, 2004.

[28] N. Vora and P. Daoutidis, "Nonlinear model reduction of chemical reaction systems," *AIChE Journal*, vol. 47, no. 10, pp. 2320–2332, 2001.

[29] S. H. Lam, "Model reductions with special csp data," *Combustion and Flame*, vol. 160, no. 12, pp. 2707 –2711, 2013.

[30] C. Brochot, J. Toth, and F. Y. Bois, "Lumping in pharmacokinetics," *J Pharmacokinet Pharmacodyn*, vol. 32, no. 5-6, pp. 719–736, 2005.

[31] J. C. Liao and E. N. Lightfoot, "Lumping analysis of biochemical reaction systems with time scale separation," *Biotechnology and Bioengineering*, vol. 31, no. 8, pp. 869–879, 1988.

[32] H. Huang, M. Fairweather, J. Griffiths, A. Tomlin, and R. Brad, "A systematic lumping approach for the reduction of comprehensive kinetic models," *Proceedings of the Combustion Institute*, vol. 30, no. 1, pp. 1309 –1316, 2005.

[33] G. Liu, M. T. Swihart, and S. Neelamegham, "Sensitivity, principal component and flux analysis applied to signal transduction: The case of epidermal growth factor mediated signaling," *Bioinformatics*, vol. 21, no. 7, pp. 1194–1202, 2005.

[34] H. Schmidt, M. F. Madsen, S. Dan, and G. Cedersund, "Complexity reduction of biochemical rate expressions," *Bioinformatics*, vol. 24, no. 6, pp. 848–854, 2008.

[35] J. Jeong, N. J. Frohberg, E. Zhou, T. Sulchek, and P. Qiu, "Accurately tracking single-cell movement trajectories in microfluidic cell sorting devices," *PLOS ONE*, vol. 13, no. 2, pp. 1–16, Feb. 2018.

[36] A. Fritsch, M. Hckel, T. Kiessling, K. D. Nnetu, F. Wetzel, M. Zink, and J. A. Ks, "Are biomechanical changes necessary for tumour progression?" *Nature Physics*, vol. 6, p. 730, 2010.

[37] G. Wang, W. Mao, R. Byler, K. Patel, C. Henegar, A. Alexeev, and T. Sulchek, "Stiffness dependent separation of cells in a microfluidic device," *PLOS ONE*, vol. 8, no. 10, pp. 1–10, Oct. 2013.

[38] B. Tasadduq, W. Lam, A. Alexeev, A. F. Sarioglu, and T. Sulcheck, "Enhancing size based size separation through vertical focus microfluidics using secondary flow in a ridged microchannel," *Scientific Reports*, vol. 7, p. 17 375, 1 2017.

[39] M. Islam, H. Brink, S. Blanche, C. DiPrete, T. Bongiorno, N. Stone, A. Liu, A. Philip, G. Wang, W. Lam, A. Alexeev, E. K. Waller, and T. Sulchek, "Microfluidic sorting of cells by viability based on differences in cell stiffness," *Scientific Reports*, vol. 7, p. 1997, 1 2017.

[40] G. Wang, K. Crawford, C. Turbyfield, W. Lam, A. Alexeev, and T. Sulchek, "Microfluidic cellular enrichment and separation through differences in viscoelastic deformation," *Lab Chip*, vol. 15, pp. 532–540, 2 2015.

[41] S. A. Kauffman, "Metabolic stability and epigenesis in randomly constructed genetic nets," *J. Theor. Biol.*, vol. 22, no. 3, pp. 437–467, 1969.

[42] S. K. Jha and C. J. Langmead, "Exploring behaviors of stochastic differential equation models of biological systems using change of measures," *BMC Bioinformatics*, vol. 13 Suppl 5, S8, 2012.

[43] W. Materi and D. S. Wishart, "Computational systems biology in drug discovery and development: methods and applications," *Drug Discov. Today*, vol. 12, no. 7-8, pp. 295–303, 2007.

[44] S. Boyd and L. Vandenberghe, *Convex Optimization*. New York, NY, USA: Cambridge University Press, 2004, ISBN: 0521833787.

[45] C. G. Moles, P. Mendes, and J. R. Banga, "Parameter estimation in biochemical pathways: A comparison of global optimization methods," *Genome Res*, vol. 13, pp. 2467–74, 11 2003.

[46] J. O. Ramsay, G. Hooker, D. Campbell, and J. Cao, "Parameter estimation for differential equations: A generalized smoothing approach," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 69, no. 5, pp. 741–796, 2007.

[47] S. Zenker, J. Rubin, and G. Clermont, "From inverse problems in mathematical physiology to quantitative differential diagnoses," *PLOS Computational Biology*, vol. 3, no. 11, pp. 1–15, Nov. 2007.

[48] D. Campbell and O. Chkrebtii, "Maximum profile likelihood estimation of differential equation parameters through model based smoothing state estimates," *Mathematical Biosciences*, vol. 246, no. 2, pp. 283 –292, 2013.

[49] M. Apri, M. de Gee, and J. Molenaar, "Complexity reduction preserving dynamical behavior of biochemical networks," *Journal of Theoretical Biology*, vol. 304, no. Supplement C, pp. 16 –26, 2012.

[50] S. Danø, M. F. Madsen, H. Schmidt, and G. Cedersund, "Reduction of a biochemical model with preservation of its basic dynamic properties," *FEBS Journal*, vol. 273, no. 21, pp. 4862–4877, 2006.

[51] P. D. Kourdis, A. G. Palasantza, and D. A. Goussis, "Algorithmic asymptotic analysis of the nf-kb signaling system," *Computers and Mathematics with Applications*, vol. 65, no. 10, pp. 1516 –1534, 2013, Grasping Complexity.

[52] A. G. Busetto, A. Hauser, G. Krummenacher, M. Sunnaker, S. Dimopoulos, C. S. Ong, J. Stelling, and J. M. Buhmann, "Near-optimal experimental design for model selection in systems biology," *Bioinformatics*, vol. 29, no. 20, pp. 2625–2632, 2013.

[53] J. Vanlier, C. A. Tiemann, P. A. J. Hilbers, and N. A. W. van Riel, "An integrated strategy for prediction uncertainty analysis," *Bioinformatics*, vol. 28, pp. 1130–5, 8 2012.

[54] E. Pauwels, C. Lajaunie, and J. P. Vert, "A bayesian active learning strategy for sequential experimental design in systems biology," *BMC Syst Biol*, vol. 8, 2014.

[55] F. Casey, D. Baird, Q. Feng, R. Gutenkunst, J. Waterfall, C. Myers, K. Brown, R. Cerione, and J. Sethna, "Optimal experimental design in an epidermal growth factor receptor signalling and down-regulation model," *IET Systems Biology*, vol. 1, no. 3, pp. 190–202, May 2007.

[56] J. C. W. Kuo and J. Wei, "Lumping analysis in monomolecular reaction systems. analysis of approximately lumpable system," *Industrial & Engineering Chemistry Fundamentals*, vol. 8, no. 1, pp. 124–133, 1969.

[57] A. Dokoumetzidis and L. Aarons, "Proper lumping in systems biology models," *IET Systems Biology*, vol. 3, 40–51(11), 1 2009.

[58] C. Seigneur, G. Stephanopoulos, and R. W. Carr, "Dynamic sensitivity analysis of chemical reaction systems: A variational method," *Chemical Engineering Science*, vol. 37, no. 6, pp. 845 –853, 1982.

[59] T. Turnyi, T. Brces, and S. Vajda, "Reaction rate analysis of complex kinetic systems," *International Journal of Chemical Kinetics*, vol. 21, no. 2, pp. 83–99, 1989.

[60] L. Petzold and W. Zhu, "Model reduction for chemical kinetics: An optimization approach," *AIChE Journal*, vol. 45, no. 4, pp. 869–886, 1999.

[61] B. Steiert, A. Raue, J. Timmer, and C. Kreutz, "Experimental design for parameter estimation of gene regulatory networks," *PLOS ONE*, vol. 7, no. 7, pp. 1–11, Jul. 2012.

[62] T. Maiwald, H. Hass, B. Steiert, J. Vanlier, R. Engesser, A. Raue, F. Kipkeew, H. H. Bock, D. Kaschek, C. Kreutz, and J. Timmer, "Driving the model to its limit: Profile likelihood based model reduction," *PLOS ONE*, vol. 11, no. 9, pp. 1–18, Sep. 2016.

[63] M. K. Transtrum and P. Qiu, "Model reduction by manifold boundaries," *Phys. Rev. Lett.*, vol. 113, p. 098 701, 9 2014.

[64] M. K. Transtrum and P. Qiu, "Bridging Mechanistic and Phenomenological Models of Complex Biological Systems," *PLoS Comput. Biol.*, vol. 12, no. 5, e1004915, May 2016.

[65] J. E. Jeong, Q. Z. an Mark K. Transtrum, E. Zhou, and P. Qiu, "Experimental design and model reduction in systems biology," *Quantitative Biology*, 0, p. 0,

[66] Z. Kutalik, K. H. Cho, and O. Wolkenhauer, "Optimal sampling time selection for parameter estimation in dynamic pathway modeling," *BioSystems*, vol. 75, no. 1-3, pp. 43–55, 2004.

[67] S. Bandara, J. P. Schloder, R. Eils, H. G. Bock, and T. Meyer, "Optimal experimental design for parameter estimation of a cell signaling model," *PLoS Comput. Biol.*, vol. 5, no. 11, e1000558, 2009.

[68] D. R. Hagen, J. K. White, and B. Tidor, "Convergence in parameters and predictions using computational experimental design," *Interface Focus*, vol. 3, no. 4, p. 20 130 008, 2013.

[69] T. Toni, D. Welch, N. Strelkowa, A. Ipsen, and M. P. Stumpf, "Approximate bayesian computation scheme for parameter inference and model selection in dynamical systems," *Journal of The Royal Society Interface*, vol. 6, no. 31, pp. 187–202, 2009.

[70] B. R. Frieden, *Physics from fisher information: A unification*, 2000.

[71] M. K. Transtrum, B. B. Machta, and J. P. Sethna, "Geometry of nonlinear least squares with applications to sloppy models and optimization," *Phys. Rev. E*, vol. 83, p. 036 701, 3 2011.

[72] J. R. Leis and M. A. Kramer, "The simultaneous solution and sensitivity analysis of systems described by ordinary differential equations," *ACM Trans. Math. Softw.*, vol. 14, no. 1, pp. 45–60, Mar. 1988.

[73] A. Kumar, P. D. Christofides, and P. Daoutidis, "Singular perturbation modeling of nonlinear processes with nonexplicit time-scale multiplicity," *Chemical Engineering Science*, vol. 53, no. 8, pp. 1491 –1504, 1998.

[74] T. J. Snowden and M. J. van der Graaf P. H.and Tindall, "Methods of model reduction for large-scale biological systems: A survey of current methods and trends," *Bull Math Biol*, vol. 79, pp. 1449–86, 7 2017.

[75] R. Heinrich and S. Schuster, *The regulation of cellular systems*. Chapman and Hall, 1996.

[76] E. Voit, *A First Course in Systems Biology*, 1st. Garland Science, 2012, ISBN: 0815344678, 9780815344674.

[77] M. S. Okino and M. L. Mavrovouniotis, "Simplification of Mathematical Models of Chemical Reaction Systems," *Chem. Rev.*, vol. 98, no. 2, pp. 391–408, 1998.

[78] J. Wolf and R. Heinrich, "Effect of cellular interaction on glycolytic oscillations in yeast: a theoretical investigation," *Biochem. J.*, vol. 345 Pt 2, pp. 321–334, 2000.

[79] H. Conzelmann, J. Saez-Rodriguez, T. Sauter, E. Bullinger, F. Allgower, and E. D. Gilles, "Reduction of mathematical models of signal transduction networks: simulation-based approach applied to EGF receptor signalling," *Syst Biol (Stevenage)*, vol. 1, no. 1, pp. 159–169, 2004.

[80] W. Liebermeister, U. Baur, and E. Klipp, "Biochemical network models simplified by balanced truncation," *FEBS J.*, vol. 272, no. 16, pp. 4034–4043, 2005.

[81] J Maertens, B. Donckels, G Lequeux, and P. Vanrolleghem, "Metabolic model reduction by metabolite pooling on the basis of dynamic phase planes and metabolite correlation analysis," in *Proceedings of the Conference on Modeling and Simulation in Biology, Medicine and Biomedical Engineering. Linkping, Sweden*, 2005.

[82] J. Jeong and P. Qiu, "The relative importance of data points in systems biology and parameter estimation," in *2017 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2017, pp. 367–373.

[83] J. E. Jeong and P. Qiu, "Quantifying the relative importance of experimental data points in parameter estimation," *BMC Systems Biology*, vol. 12, no. 6, p. 103, 2018.

[84] F. A. Potra and S. J. Wright, "Interior-point methods," *Journal of Computational and Applied Mathematics*, vol. 124, pp. 281–302, 2000.

[85] D. van Ravenzwaaij, P. Cassey, and S. D. Brown, "A simple introduction to markov chain monte–carlo sampling," *Psychonomic Bulletin & Review*, vol. 124, pp. 281–302, 2016.

[86] Z. Deng and T. Tian, "A continuous optimization approach for inferring parameters in mathematical models of regulatory networks," *BMC Bioinformatics*, vol. 15, no. 1, p. 256, 2014.

[87] M Swat, A Kel, and H Herzel, "Bifurcation analysis of the regulatory modules of the mammalian g1/s transition," *Bioinformatics*, vol. 20, pp. 1506–11, 2004.

[88] E. Voit, *A First Course in Systems Biology*, 1st. New York: Garland Science, 2012, pp. 254–255, ISBN: 0815344678, 9780815344674.

[89] F. Hoppensteadt, "Predator-prey model," *Scholarpedia*, vol. 1, no. 10, p. 1563, 2006, revision #91667.

[90] A. Gábor and J. R. Banga, "Robust and efficient parameter estimation in dynamic models of biological systems," *BMC Systems Biology*, vol. 9, no. 1, p. 74, 2015.

[91] F. Bauer and M. A. Lukas, "Comparingparameter choice methods for regularization of ill-posed problems," *Mathematics and Computers in Simulation*, vol. 81, no. 9, pp. 1795 –1841, 2011.

[92] C. Shields, C. Reyes, and G. Lpez, "Microfluidic cell sorting: A review of the advances in the separation of cells from debulking to rare cell isolation," *Lab Chip*, vol. 15, pp. 1230–49, 5 2015.

[93] T. W. Murphy, Q. Zhang, L. B. Naler, S. Ma, and C. Lu, "Recent advances in the use of microfluidic technologies for single cell analysis," *Analyst*, vol. 143, pp. 60–80, 1 2018.

[94] Y. Zheng, J. Nguyen, Y. Wei, and Y. Sun, "Recent advances in microfluidic techniques for single-cell biophysical characterization," *Lab on a Chip*, vol. 13, no. 13, pp. 2464–2483, 2013.

[95] A. Vinckier and G. Semenza, "Measuring elasticity of biological materials by atomic force microscopy," *FEBS Letters*, vol. 430, no. 1, pp. 12 –16, 1998.